HU ISSN 1586–2070

JOURNAL OF COMPUTATIONAL AND APPLIED MECHANICS

A Publication of the University of Miskolc

VOLUME 4, NUMBER 1 (2003)

MISKOLC UNIVERSITY PRESS

HU ISSN 1586–2070

JOURNAL OF COMPUTATIONAL AND APPLIED MECHANICS

A Publication of the University of Miskolc

VOLUME 4, NUMBER 1 (2003)

MISKOLC UNIVERSITY PRESS

EDITORIAL BOARD

LOCAL EDITORIAL COUNCIL

Journal of Computational and Applied Mechanics, Vol. 4., No. 1., (2003), pp. 3–4

PREFACE

The fourth volume contains papers on solid and fluid mechanics selected from those presented at the ninth International Conference on Numerical Mathematics and Computational Mechanics (NMCM2002) held 15-19 July 2002 at the University of Miskolc, Hungary.

This conference was the latest in a series of international conferences initiated by Hungarian mathematicians under the leadership of Professor Pál Rózsa who organized and chaired the first six conferences.

Initially the conferences focused on the theory and applications of differential equations and numerical algebra. Their scope gradually widened to include, for the first time in 1994, computational mechanics. The seventh conference (NMCM96), chaired by Professor Ivo Babuška, was a satellite conference to the 2nd European Congress of Mathematics held in Budapest July 21-27, 1996. The conference emphasized the interdisciplinary nature of numerical methods and computational mechanics and the importance of mathematical concepts in the development of numerical simulation methods to achieve quality, reliability and efficiency in numerical simulation.

Selected papers from NMCM96 were published in special editions of three prestigious journals: Computer Methods in Applied Mechanics and Engineering (guest editor: Ivo Babuška); Computers and Mathematics with Applications (guest editors: B. Szabó, A. Galántai and G. Szeidl) and Computer Assisted Mechanics and Engineering Sciences (guest editors: B. Szabó, A. Galántai and G. Szeidl).

The goal of NMCM98 was similar to that of NMCM96: Provide a forum to researchers working in the interrelated fields of numerical methods and computational mechanics to discuss recent developments, current problems and challenges and explore new approaches to interdisciplinary problem solving in numerical simulation.

Selected papers from NMCM98 were also published in special editions of the Computers and Mathematics with Applications (guest editors: B. Szabó, A. Galántai, K. Balla and G. Szeidl) and Computer Assisted Mechanics and Engineering Sciences (guest editors: B. Szabó, A. Galántai and G. Szeidl).

The goal of NMCM2002 is similar to that of the previous conferences with the additional objective to provide a forum for young researchers from various countries of the European Community and neighbouring countries.

Our special thanks to Prof. Vladimir Kompiš of the University of Zilina, Slovak Republic, for including this conference in the series of Euro Conferences that he organized.

We are particularly grateful to Professors Olaf Axelson, Ulrich Gabbert, Robert Haber, Vladimir Kompiš, Herbert Mang, Rosvita März, Anna Sändig, Jan Sladek, Endre Süli, Wolfgang Wendland, Taifun Tezduyar, Tibor Czibere, Zsolt Gáspár, Imre Kozák, István Páczelt, Pál Rózsa and Gisbert Stoyan for their service on the International Science Committee and for having provided advice and support in the course of preparations.

We would like to thank the sponsors. First of all the European Commission for their generous support provided for young researchers, the keynote speakers and the members of the International Science Committee. The University of Miskolc, the Hungarian Academy of Sciences, the Hungarian Committee of Technological Development (OMFB), the Foundation for the University of Miskolc and the Hungarian Credit Bank Foundation for the Hungarian Technological Development.

Barna Szabó Chairman
Katalin Balla
Aurél Galantai
György Szeidl
Editors of the Fourth Volume

Journal of Computational and Applied Mechanics, Vol. 4., No. 1., (2003), pp. 5–12

PARALLELIZATION OF AN ALGORITHM FOR SOLVING THE GRAVITY INVERSE PROBLEM

Elena N. Akimova
Institute of Mathematics and Mechanics, Urals Branch of the Russian Academy of Sciences
S. Kovalevskaya Str., 16, Ekaterinburg, 620219, Russia
aen@imm.uran.ru

[Received: August 14, 2002]

Abstract. A parallel algorithm for solving the gravity inverse problem is considered. The corresponding programme has been implemented on the Massively Parallel Computing System MVS–1000.

Mathematical Subject Classification: 65F05, 86A22
Keywords: gravity inverse problem, parallel algorithm

1. Introduction

The three-dimensional gravity inverse problem of finding the interface between mediums from the gravitational data is investigated. A model of the lower half-space consists of the three mediums with constant densities which are divided by the surfaces S_1 and S_2. The gravitational anomaly is formed by the deviation of the desired surface S from the horizontal plane $z = H$ ($H_1 = 2,\ H_2 = 10$ in our case) [1].

2. Main equations and numerical algorithms

The gravity equation with respect to the unknown surface $z = z(x, y)$ is reduced to the two-dimensional nonlinear equation with integral operator

$$B[z] \equiv f\Delta\sigma \int_a^b \int_c^d \left\{ \frac{1}{[(x-x')^2 + (y-y')^2 + z^2(x',y')]^{1/2}} - \right.$$
$$\left. - \frac{1}{[(x-x')^2 + (y-y')^2 + H^2]^{1/2}} \right\} dx'dy' = F(x,y), \quad (2.1)$$

where f is the gravitation constant, $\Delta\sigma$ is the density jump on the interface, $F(x, y)$ is the anomalous gravitational field.

For solving the nonlinear integral equation the iterative regularizing Newton method is used

$$z^{k+1} = z^k - [B'(z^k) + \alpha_k I]^{-1}(B(z^k) + \alpha_k z^k - F), \quad (2.2)$$

where $B'(z^k)$ is the Frechet derivative of the operator B in the point z^k for (2.1), I is the identity operator, α_k is a sequence of the positive parameters which are chosen taking into account the right side of equation (2.1).

After discretizing equation (2.1) on the grid $n = M \times N$ and approximating the integral operator B by the quadrature formulas, the problem of the form (2.2) can be written in the form of the system of linear equations with asymmetric and full $n \times n$ matrix and can be solved by the Gauss or Gauss-Jordan elimination algorithms for each iteration of the method (2.2).

We consider the system of n linear equations with n unknowns in the form:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = a_{1n+1} \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n = a_{2n+1} \\ ..... \\ a_{n1}x_1 + a_{n2}x_2 + ... + a_{nn}x_n = a_{nn+1} \end{cases} . \tag{2.3}$$

The main idea of the Gaussian elimination method is to reduce the full matrix A of the system (2.3) to the upper triangular form, that is to obtain the system of equations in the following form:

$$\begin{cases} x_1 + c_{12}x_2 + c_{13}x_3 + ... + c_{1n}x_n = c_{1n+1} \\ x_2 + c_{23}x_3 + ... + c_{2n}x_n = c_{2n+1} \\ ..... \\ x_n = c_{nn+1} \end{cases} . \tag{2.4}$$

From the system (2.4) we find the unknowns by the formulas:

$$x_k = c_{kn+1} - c_{kk+1}x_{k+1} - ... - c_{kn}x_n, \quad k = n, n-1, ..., 1 . \tag{2.5}$$

In the first step of the Gauss elimination method we choose the pivot element (the maximum of modulus) in the first equation of the system (2.3). Let $a_{11} \neq 0$. After dividing all the coefficients and the constant term of the first equation of the system by a_{11} we obtain the following equation:

$$x_1 + c_{12}x_2 + c_{13}x_3 + ... + c_{1n}x_n = c_{1n+1}, \tag{2.6}$$

where $c_{1j} = a_{1j}/a_{11}, \quad j = 2, 3, ..., n+1.$

With the help of equation (2.6) we eliminate the unknown x_1 from the other equations of the system, beginning with the second one. We get the following system:

$$\begin{cases} a^{(1)}_{22}x_2 + a^{(1)}_{23}x_3 + ... + a^{(1)}_{2n}x_n = a^{(1)}_{2n+1} \\ a^{(1)}_{32}x_2 + a^{(1)}_{33}x_3 + ... + a^{(1)}_{3n}x_n = a^{(1)}_{3n+1} \\ ..... \\ a^{(1)}_{n2}x_2 + a^{(1)}_{n3}x_3 + ... + a^{(1)}_{nn}x_n = a^{(1)}_{nn+1} \end{cases} \tag{2.7}$$

where $a^{(1)}_{ij} = a_{ij} - c_{1j}a_{i1}, \quad i = 2, 3, ..., n, \quad j = 2, 3, ..., n+1.$

Continuing the process, in the k-th step we obtain the equation

$$x_k + c_{kk+1}x_{k+1} + ... + c_{kn}x_n = c_{kn+1}, \tag{2.8}$$

where $c_{kj} = a_{kj}^{(k-1)}/a_{kk}^{(k-1)}$, $j = k+1, ..., n+1$, and the system of equations

$$\begin{cases} a_{k+1k+1}^{(k)} x_{k+1} + ... + a_{k+1n}^{(k)} x_n = a_{k+1k+1}^{(k)} \\ ..... \\ a_{nk+1}^{(k)} x_{k+1} + ... + a_{nn}^{(k)} x_n = a_{nn+1}^{(k)} \end{cases}, \tag{2.9}$$

where $a_{ij}^{(k)} = a_{ij}^{(k-1)} - c_{kj} a_{ik}^{(k-1)}$, $i = k+1, ..., n$, $j = k+1, ..., n+1$.

In the last step of the elimination we have the equation $x_n = c_{nn+1}$.

The Gauss-Jordan algorithm is one of the variants of the Gaussian elimination algorithm. In this case the matrix A of system (2.3) is reduced to diagonal form, but not an upper triangular form. In the $(k+1)$-th step the current matrix A_k has the following form:

$$A_k = \begin{bmatrix} 1 & 0 & \dots & 0 & a_{1,k+1}^{(k)} & \dots & a_{1n}^{(k)} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & a_{k,k+1}^{(k)} & \dots & a_{kn}^{(k)} \\ 0 & 0 & \dots & 0 & a_{k+1,k+1}^{(k)} & \dots & a_{k+1,n}^{(k)} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & a_{n,k+1}^{(k)} & \dots & a_{nn}^{(k)} \end{bmatrix}. \tag{2.10}$$

We divide the $(k+1)$-th row by the coefficient $a_{k+1,k+1}^{(k)}$ and eliminate all off-diagonal elements of the $(k+1)$-th column. We will make this elimination by multiplying the $(k+1)$-th row by $a_{j,k+1}^{(k)}$ and subtracting the result from the j-th row ($j=1, ..., n$; $j \neq k+1$).

To guarantee the numerical stability of the Gauss and Gauss-Jordan algorithms in the general case, a partial choice of the pivot element is necessary. If we take the maximum (with respect to the modulus) element in the k-th row as the pivot element in the k-th step of the elimination, then before the realization of the modification it is necessary to rearrange the k-th column and the column with the pivot element.

3. **Parallel realization**

The parallel realization of the Gauss method for m processors is the following. Conditionally, we divide the vectors z and F into m parts so that $n = m \cdot L$. The matrix A is divided by the horizontal lines into the m blocks, respectively (Figure 1). Assume that the rows of the matrix A with numbers $1, 2, ..., L$ are stored in the memory of the first processor (the Host), the rows with numbers $(L+1), (L+2), ..., 2L$ are stored in the memory of the second processor, the rows with numbers $(2L+1), (2L+2), ..., 3L$ are stored in the memory of the third processor, and so on. The rows with numbers $(m-1)L+1, ..., Lm$ are stored in the memory of the m-th processor. In the first step the Host processor chooses the pivot element among the elements $a_{11}, a_{12}, ..., a_{1n}$ of the first row, modifies the first row and sends it to each of the other processors. After that each processor eliminates the first unknown x_1 of every row from its own part of the L equations. After the first step of the elimination we obtain the matrix with the first column $(1, 0, 0, ..., 0)^T$. In the second step the Host processor sends the modified

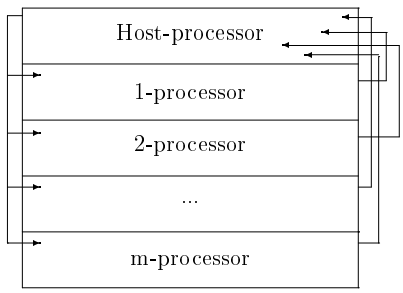


Figure 1. The diagram of the data distribution over the processors

second row to each of the other processors. After that each processor eliminates the second unknown x_2 of every row from its own part of the $(L-1)$ equations, and so on, up to the L-th step. At the end of the L-th step the second processor sends the L-th row to the Host processor. The Host chooses the pivot element, modifies the L-th row and sends it to each of the other processors. After that each processor of the others eliminates the unknown x_{L+1} of every row from its own part of the equations. At the end of the $2L$-th step the third processor sends the $(2L+1)$-th row to the Host processor. The Host chooses the pivot element, modifies the $(2L+1)$-th row and sends it to each of the other processors, and so on. In the last step the Host processor sends the modified last row to each of the other processors.

During the elimination process more and more processors become idle in every step, since the number of the equations is diminished by one. This affects the efficiency of the algorithm. The Host processor works until the end because it responds to the transfer of the modified rows and the choice of the pivot element in every step. To reduce the waiting time the Host processor sends the modified row to other processors immediately after receiving it and makes the calculations with its own part of the equations independently. In the Gauss-Jordan method all the processors make calculations with their own parts until the end. The waiting time decreases and the efficiency of the algorithm increases.

4. **Efficiency**

Parallelization of the basic algorithms and their realization on the Massively Parallel Computing System MVS–1000 [2] are implemented. The analysis of the efficiency of parallelization of the iterative algorithm with different numbers of processors is carried out. MVS–1000/16 of the Research Institute is a KVANT computer consisting of 16 Intel Pentium III-800, 256 MByte, 10 GByte disk, two 100 Mbit network controllers (Digital DS21143 Tulip and Intel PRO/100). The educational computing cluster consists of 8 Intel Pentium III-700, 128 MByte, 14 GByte disk, 100 Mbit network controller 3Com 3c905B Cyclone (Figure 2). The first 15 nodes work fast, the other 8 nodes work more slowly.

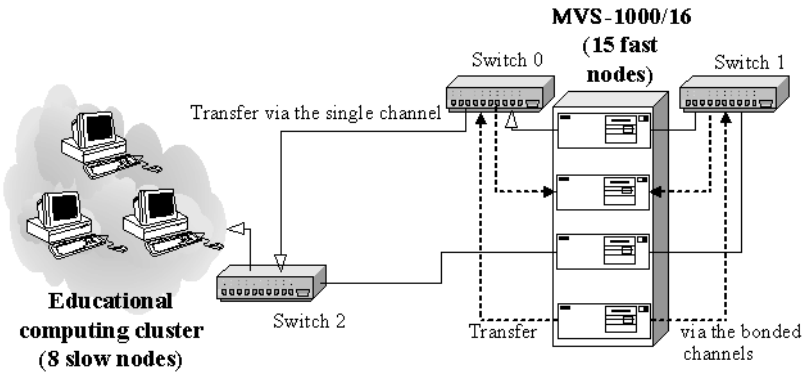


Figure 2. MVS–1000/16

For comparison of the execution times of the sequential and parallel algorithms, we will consider the coefficients of the speed-up and efficiency

$$S_m = T_1/T_m, \qquad E_m = S_m/m,$$

where T_m is the execution time of the parallel algorithm on MVS–1000 with m ($m > 1$) processors, T_1 is the execution time of the sequential algorithm on one processor.

$T_m = T_c + T_o$, where T_c is the computing time, T_o is the exchange time. The number m of processors corresponds to the division of the vectors z and F into m parts mentioned so that $n = m \cdot L$.

Table 1 shows execution times and the coefficients of the speed-up and the efficiency of the iterative regularizing Newton method with 5 iterations with using the parallel and sequential ($m = 1$) Gauss algorithms for problem (2.1)— (2.2) for 111×35 points of the grid domain.

m	T_m, min.	S_m	E_m
1	57.48	—	—
2	46.85	1.23	0.61
3	36.18	1.59	0.53
4	29.38	1.96	0.49
5	25.78	2.23	0.45
6	21.83	2.63	0.44
8	17.25	3.33	0.42
10	16.17	3.55	0.36
12	15.32	3.75	0.31

Table 1. Gauss Method

m	T_m, min.	S_m	E_m
1	114.1	—	—
2	60.50	1.89	0.94
3	42.38	2.69	0.90
4	33.53	3.40	0.85
5	28.48	4.01	0.80
6	23.88	4.78	0.79
8	19.88	5.74	0.72
10	18.45	6.18	0.62
12	17.35	6.58	0.55

Table 2. Gauss-Jordan Method

Table 2 shows execution times and the coefficients of the speed-up and the efficiency of the iterative regularizing Newton method with 5 iterations with using the parallel and sequential ($m = 1$) Gauss-Jordan algorithms for problem (2.1)–(2.2) for 111×35 points of the grid domain.

The results of the calculations show that the parallel Gauss and Gauss-Jordan algorithms have quite a high efficiency of parallelization, and the Gauss-Jordan algorithm efficiency is higher than the efficiency of the Gauss algorithm. In the case of the parallel Gauss algorithm with the number of processors $m \leq 5$, the efficiency is $E_m > 0.45$. In the case of the parallel Gauss-Jordan algorithm with the number of processors $m \leq 10$, the efficiency is $E_m > 0.6$. When the number of processors m is small, then the speed-up S_m increases almost linearly as the number m increases. On the other hand, when m is large, then the exchange time increases, so the efficiency E_m decreases.

5. **Concluding remarks**

The preliminary gravitational data processing is connected with the selection of the anomalous field for each interface S_i $(i = 1, 2)$ from the common data measured on a rectangular area in some region in the Urals. This processing was implemented by the methods from [1]. In Figures 3 and 4 the graph 1 of the boundary profiles of the interfaces S_1 and S_2 is obtained by using the methods from [1] (continuous lines). Graph 2 of the boundary profiles is obtained by solving the problem (2.1) by the iterative regularizing Newton method (2.2) with the aid of the parallel Gauss or Gauss-Jordan algorithms (dotted lines).

For approximation of the integral operator (2.2) we used the two-dimensional analog of the rectangle quadrature formulas for 111×35 points of the grid domain with grid widths $h_x = 0.5$ and $h_y = 2$ (km). The positive parameters α_k were chosen from numerical experiments taking into account the right side of equation (2.1), namely $\alpha_k \ll F(x, y)$. In Figure 3 for problem (2.1) $H = 2$ (km) is the depth of the surface

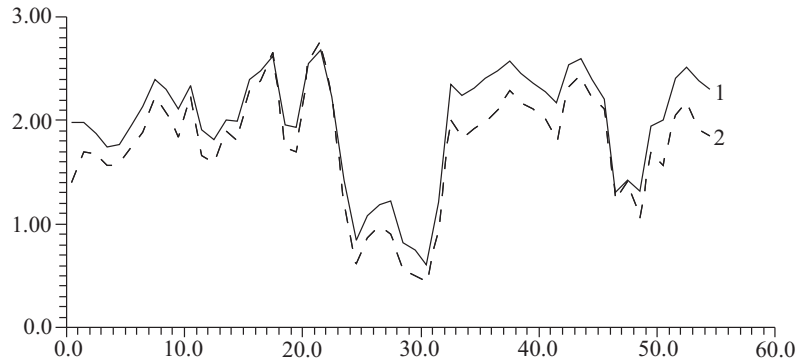


Figure 3. Boundary profiles for $H = 2$ (km)

S, $f = 6.67 \cdot 10^{-5}$ is the gravitation constant, $\Delta\sigma = 0.48$ (g/cm^3) is the density jump on the interface, $z_0(x, y) = 0.3$ (km) is the initial approximation and $\alpha_k = 2.5$ is a sequence of the positive parameters. In Figure 4 for problem (2.1) $H = 10$ (km) is the

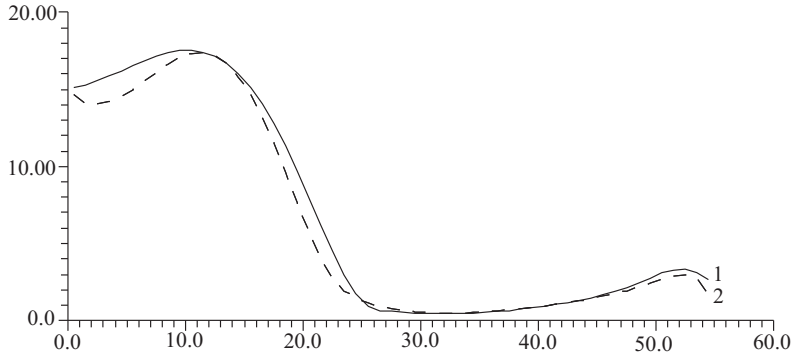


Figure 4. Boundary profiles for $H = 10$ (km)

depth of the surface S, $f = 6.67 \cdot 10^{-5}$ is the gravitation constant, $\Delta\sigma = 0.23$ (g/cm^3) is the density jump on the interface, $z_0(x, y) = 0.3$ (km) is the initial approximation and $\alpha_k = 0.8$ is a sequence of the positive parameters.

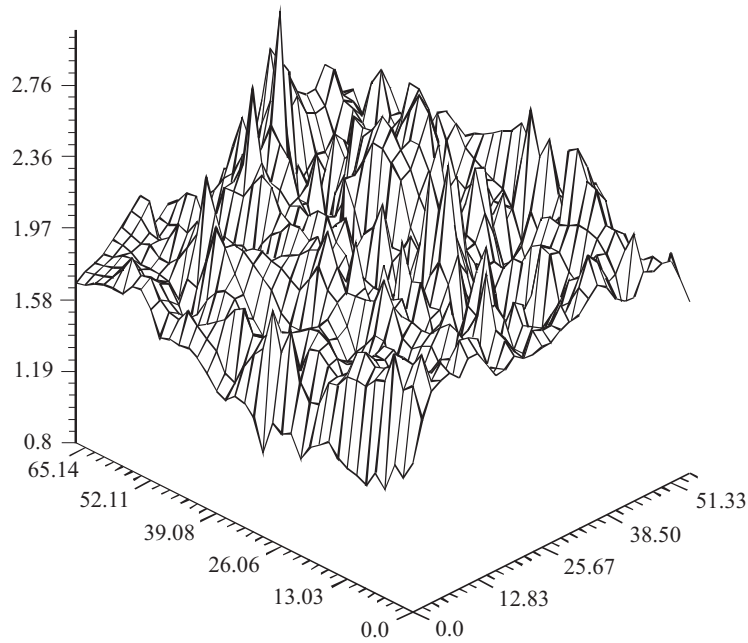


Figure 5. The reconstructed interface S_1

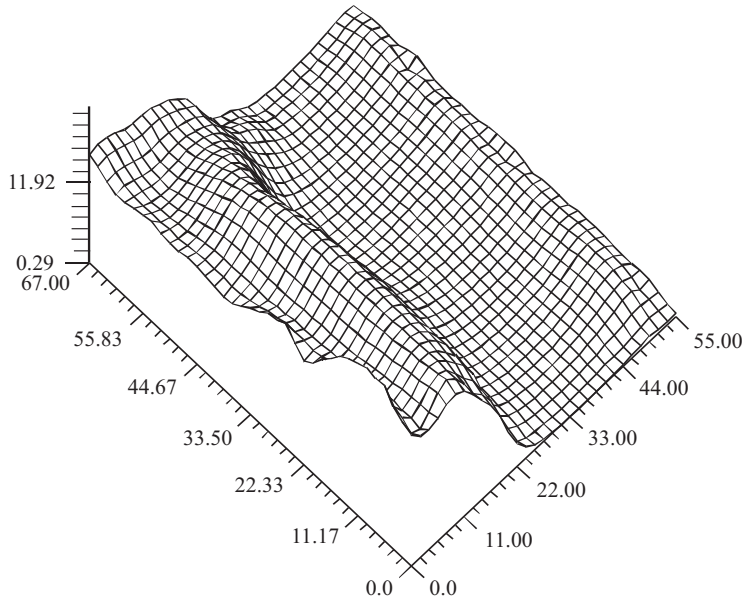


Figure 6. The reconstructed interface S_2

In Figures 5 and 6 the three-dimensional interfaces S_1 and S_2 for the real gravity field of some area in the Urals for $H = 2$ (km) and $H = 10$ (km) are represented. They are reconstructed by the iterative regularizing Newton method (2.2) with the help of the parallel Gauss or Gauss-Jordan algorithms.

The main conclusion is the following. The interfaces S_1 and S_2 obtained as solutions of the gravity inverse problem (2.1) by the iterative regularizing Newton method (2.2) correspond to the real conceptions about the investigated region in the Urals. Parallelization of the algorithms decreases the time of solving the problems.

Acknowledgement. The support provided by the Russian Foundation for Basic Research (project No. 03-01-00099) is gratefully acknowledged. The author is grateful to V.V. Vasin, the corresponding member of the Russian Academy of Sciences for the formulation of the problem and attention to the work.

References

1. Vasin, V.V., Perestoronina, G.Ya., Prutkin, I.L. and Timerkhanova, L.Yu.: *Reconstruction of relief of geological boundaries in the three-layered medium using the gravitational and magnetic data.* In: *Proceedings of "Geophysics and Mathematics",* 35-41. Institute of Mines, UrB RAS, Perm, 2001.
2. Baranov, A.V., Latsis, A.O., Sazhin, C.V. and Khramtsov, M.Yu. The MVS–1000 System User's Guide. *http://parallel.ru/mvs/user.html.*

Journal of Computational and Applied Mechanics, Vol. 4., No. 1., (2003), pp. 13–25

COMPUTATION OF UNSTEADY MOMENTUM AND HEAT TRANSFER FROM A FIXED CIRCULAR CYLINDER IN LAMINAR FLOW

László Baranyi
Department of Fuid and Heat Engineering, University of Miskolc
3515 Miskolc - Egyetemváros, Hungary
arambl@gold.uni-miskolc.hu

[Received: December 20, 2002]

Abstract. This paper presents a finite difference solution for 2D, low Reynolds number, unsteady flow around and heat transfer from a stationary circular cylinder placed in a uniform flow. The fluid is assumed to be incompressible and of constant property. The governing equations are the Navier-Stokes equations, the continuity equation, a Poisson equation for pressure and the energy equation. The temperature of the cylinder wall is kept constant and the viscous energy dissipation term is neglected in the energy equation. The computed Strouhal numbers, time-mean drag and base pressure coefficients, as well as the average Nusselt numbers compare well with existing experimental results.

Mathematical Subject Classification: 73A05
Keywords: forced convection, circular cylinder, Navier-Stokes equations, Finite Difference Method

1. Introduction

Investigating flow around a single circular cylinder has been the objective of a huge number of researchers, through experimental, theoretical, and numerical approaches to the problem. Despite the simple geometry, the problem is extremely complex and remains an intriguing one, and one with many applications. Knowledge of flow patterns around bluff bodies is important in the design of structures such as smokestacks, skyscrapers, or bridges. Long slender structures in wind are often subjected to large amplitude oscillation due to alternating vortex shedding, sometimes causing collapse of the structure. The same vortex shedding causes noisy operation of heat exchangers, as the pipes of the heat exchanger vibrate. These are some of the hydrodynamic aspects; another aspect of importance is the heat transfer to and from the fluid. Heat exchangers, hot wire anemometers, and cooling towers are some examples in which heat transfer is central; heat transfer between structures and the outside air is also affected by vortex shedding.

Because of the practical importance of this problem, there are numerous studies dealing with flow past cylinders that are fixed, oscillating, rotating, or in orbital motion. Among these, the fixed cylinder is usually the starting point of investigations, because it is relatively simple to carry out experiments on, and thus also for numerical studies, comparison with experimental data is possible to confirm validity of the computer method used. Roshko [1], Norberg [2], and Bearman [3] are among those who have provided invaluable experimental data on flow around a fixed cylinder. Computations on the same problem have been performed by many researchers, including Kawamura and Kuwahara [4], Braza et al. [5], and Karniadakis and Triantafyllou [6].

Heat transfer has been investigated for fixed heated cylinders in a uniform stream at low Reynolds numbers with a focus on application to hot wire anemometry by investigators such as Lange et al. [7]. In their case, computations were carried out at extremely low Reynolds numbers as well, and they claim to have identified the critical Reynolds number where vortex shedding starts at Re=45.9.

Mahfouz and Badr [8] carried out a numerical study on a fixed and a rotationally oscillating cylinder between Re=40 and 200. They attempted to investigate the possibility of controlling heat transfer using a rotationally oscillating cylinder, and found that heat transfer was appreciably enhanced in the lock-in range, and the effect of the oscillation on heat transfer was insignificant for non-lock-in cases.

In the author's earlier studies, computations were carried out for the flow around a fixed circular cylinder at different Reynolds numbers, from Re=10 to 1000, and up to Re=180 for an oscillating cylinder (e.g. Baranyi and Shirakashi, [9]), and good agreement was obtained with experimental data for the variation of Strouhal number and time mean drag coefficient with Reynolds number. Here, further features of flow are investigated for a fixed cylinder and compared with experimental data. The base pressure coefficient, which influences the near-wake structure, is investigated and compared with the experimental data of Roshko [10]. Also, the author's computational results were compared with those of Sherwin's unpublished computations carried out by a different method, for the time mean drag coefficient, Strouhal number, and the root-mean square (rms) values of lift and drag coefficients at several Reynolds numbers up to Re=140. These comparisons were used to further validate the computer method used. Based on the results, the method was extended to investigate the heat transfer from a heated cylinder in which the temperature is kept constant and the fluid is assumed to be constant property incompressible fluid.

2. **Governing equations**

When deriving the basic equations constant property incompressible fluid flow is assumed. The governing equations in non-dimensional forms are the two components of the Navier-Stokes equations, the equation of continuity, a Poisson equation for pressure p, and the energy equation:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{\mathrm{Re}}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right); \tag{2.1}$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=-\frac{\partial p}{\partial y}+\frac{1}{\mathrm{Re}}\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right); \tag{2.2}$$

$$D=\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0; \tag{2.3}$$

$$\nabla^2 p=\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2}=2\left[\frac{\partial u}{\partial x}\frac{\partial v}{\partial y}-\frac{\partial u}{\partial y}\frac{\partial v}{\partial x}\right]-\frac{\partial D}{\partial t}; \tag{2.4}$$

$$\frac{\partial T}{\partial t}+u\frac{\partial T}{\partial x}+v\frac{\partial T}{\partial y}=\frac{1}{\mathrm{RePr}}\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}\right)+\frac{E}{\mathrm{Re}}\Phi. \tag{2.5}$$

In these equations quantities are made dimensionless by using a length scale L chosen to be the diameter of the cylinder d, velocity scale U which is the free stream velocity, density ρ, kinematic viscosity υ, specific heat at constant pressure c_p, and the temperature difference between the cylinder surface (w) and the outer boundary (∞) of the computational domain $\left(\tilde{T}_w-\tilde{T}_\infty\right)$. In these equations x,y are Cartesian coordinates, u,v are velocities in x,y directions, p is pressure, D is dilation, T is dimensionless fluid temperature defined by $T=\left(\tilde{T}-\tilde{T}_\infty\right)/\left(\tilde{T}_w-\tilde{T}_\infty\right)$, where $\tilde{T}$ is the dimensional temperature, Re is the Reynolds number defined by $Re=Ud/\upsilon$, Pr is the Prandtl number $Pr=\rho\upsilon c_p/k$, where k is the thermal conductivity of the fluid, E is Eckert number defined by $E=U^2/\left[c_p\left(\tilde{T}_w-\tilde{T}_\infty\right)\right]$. The gravity force is included in the pressure terms in equations (2.1) and (2.2). Φ in equation (2.5) is the viscous energy dissipation function

$$\Phi=2\left[\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial v}{\partial y}\right)^2\right]+\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)^2. \tag{2.6}$$

At low Reynolds numbers, as in this study, the viscous dissipation function can be neglected. Although the dilation D is zero by continuity (2.3), it is advisable to retain the term $\partial D/\partial t$ to avoid computational instability (Harlow and Welch, [11]).

Boundary conditions:

(R_1: cylinder surface)

$$u=v=0\ ;$$

$$\frac{\partial p}{\partial n}=\frac{1}{Re}\nabla^2 v_n;$$

$$T_1=1.$$

(R_2: undisturbed domain)

$$u=u_{pot};\quad v=v_{pot};$$

$$\frac{\partial p}{\partial n}=\left(\frac{\partial p}{\partial n}\right)_{pot};$$

$$T_2=0.$$

Here subscript *pot* stands for potential flow and n denotes the outer normal along the cylinder.

The heat transfer rate per unit area from the cylinder wall to the fluid can be obtained from the temperature distribution and may be described as

$$\dot{q}_w = -k_w \left(\frac{\partial \tilde{T}}{\partial \tilde{r}} \right)_w , \tag{2.7}$$

where $\tilde{T}$ and $\tilde{r}$ are the dimensional temperature and radius, k_w is the thermal conductivity of the fluid at cylinder temperature, and the subscript w indicates that the temperature gradient in the radial direction is evaluated at the cylinder wall. In this study constant property fluid is considered so the thermal conductivity of the fluid k is assumed to be constant, hence $k \cong k_w$. Engineers and technicians working in this field need an expression based on measurable quantities such as

$$\dot{q}_w = h \left(\tilde{T}_w - \tilde{T}_\infty \right) , \tag{2.8}$$

where h is the heat transfer coefficient. By introducing dimensionless quantities the Nusselt number Nu can be obtained as

$$Nu = \frac{hd}{k} = -\left(\frac{\partial T}{\partial R} \right)_w , \tag{2.9}$$

where T and R are the dimensionless temperature and radius. The temperature gradient in the fluid should be evaluated on the cylinder surface.

3. Transformation from the physical plane to the computational plane

We use a boundary-fitted coordinate system, hence boundary conditions can be imposed accurately. In this way interpolation leading to poor solutions can be avoided.

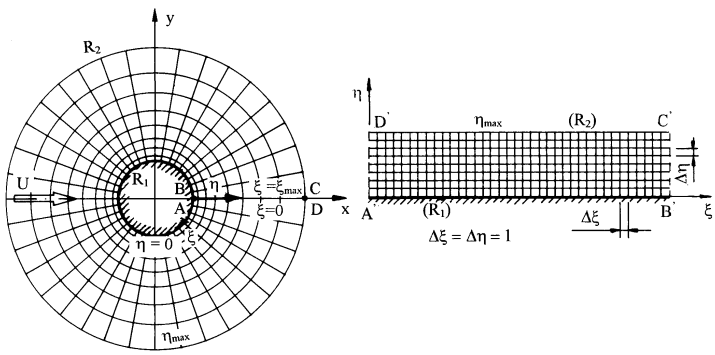


Figure 1. Physical and computational planes

A unique, single-valued relationship between the coordinates on the computational domain (ξ , η, τ) and the physical coordinates (x, y, t) (see Figure 1) can be written

as

$$x(\xi,\eta) = R(\eta)\,\cos\left[g(\xi)\right]; \quad y(\xi,\eta) = -R(\eta)\,\sin\left[g(\xi)\right]; \quad t = \tau\,, \tag{2.10}$$

where the dimensionless radius

$$R(\eta) = R_1 \exp\left[f(\eta)\right]. \tag{2.11}$$

This choice of the structure of the mapping function automatically assures that the obtained grid is orthogonal on the physical plane for arbitrary functions $f(\eta)$ and $g(\xi)$. In this study the following linear mapping functions were used

$$g(\xi) = 2\pi\frac{\xi}{\xi_{\max}}; \quad f(\eta) = \frac{\eta}{\eta_{\max}}\log\left(\frac{R_2}{R_1}\right)\,, \tag{2.12}$$

where subscript max refers to maximum value. Using mapping functions (2.12) cylindrical coordinates with logarithmically spaced radial cells are obtained on the physical plane, providing a fine grid scale near the cylinder wall and a coarse grid in the far field. Transformations (2.10)-(2.12) are single valued since in this case Jacobian J

$$J = y_\eta x_\xi - y_\xi x_\eta = \frac{2\pi\log\left(\frac{R_2}{R_1}\right)}{\xi_{\max}\eta_{\max}}R(\eta) \tag{2.13}$$

is non-vanishing, positive for an arbitrary value of η in the computational domain. In equation (2.13) subscripts ξ and η denote differentiation. After having chosen the transformations between the physical and computational domains, the governing equations can also be transformed by variables used in the computational domain.

The x and y components of the transformed Navier-Stokes equations are

$$\begin{aligned}
&\frac{\partial u}{\partial\tau} + \frac{1}{J}\left(u\frac{\partial y}{\partial\eta} - v\frac{\partial x}{\partial\eta}\right)\frac{\partial u}{\partial\xi} + \frac{1}{J}\left(v\frac{\partial x}{\partial\xi} - u\frac{\partial y}{\partial\xi}\right)\frac{\partial u}{\partial\eta} = \\
&-\frac{1}{J}\left(\frac{\partial y}{\partial\eta}\frac{\partial p}{\partial\xi} - \frac{\partial y}{\partial\xi}\frac{\partial p}{\partial\eta}\right) + \frac{1}{ReJ^2}\left(g_{22}\frac{\partial^2 u}{\partial\xi^2} + g_{11}\frac{\partial^2 u}{\partial\eta^2}\right);
\end{aligned} \tag{2.14}$$

$$\begin{aligned}
&\frac{\partial v}{\partial\tau} + \frac{1}{J}\left(u\frac{\partial y}{\partial\eta} - v\frac{\partial x}{\partial\eta}\right)\frac{\partial v}{\partial\xi} + \frac{1}{J}\left(v\frac{\partial x}{\partial\xi} - u\frac{\partial y}{\partial\xi}\right)\frac{\partial v}{\partial\eta} = \\
&-\frac{1}{J}\left(\frac{\partial x}{\partial\xi}\frac{\partial p}{\partial\eta} - \frac{\partial x}{\partial\eta}\frac{\partial p}{\partial\xi}\right) + \frac{1}{ReJ^2}\left(g_{22}\frac{\partial^2 v}{\partial\xi^2} + g_{11}\frac{\partial^2 v}{\partial\eta^2}\right).
\end{aligned} \tag{2.15}$$

Dilation D transforms as

$$D = \frac{1}{J}\left(\frac{\partial y}{\partial\eta}\frac{\partial u}{\partial\xi} - \frac{\partial y}{\partial\xi}\frac{\partial u}{\partial\eta} + \frac{\partial x}{\partial\xi}\frac{\partial v}{\partial\eta} - \frac{\partial x}{\partial\eta}\frac{\partial v}{\partial\xi}\right) = 0. \tag{2.16}$$

The Poisson equation for pressure will have the form

$$g_{22}\frac{\partial^2 p}{\partial\xi^2} + g_{11}\frac{\partial^2 p}{\partial\eta^2} = 2J\left(\frac{\partial u}{\partial\xi}\frac{\partial v}{\partial\eta} - \frac{\partial u}{\partial\eta}\frac{\partial v}{\partial\xi}\right) - J^2\frac{\partial D}{\partial\tau}. \tag{2.17}$$

The energy equation by neglecting the viscous dissipation transforms as

$$\frac{\partial T}{\partial\tau} + \frac{1}{J}\left(u\frac{\partial y}{\partial\eta} - v\frac{\partial x}{\partial\eta}\right)\frac{\partial T}{\partial\xi} + \frac{1}{J}\left(v\frac{\partial x}{\partial\xi} - u\frac{\partial y}{\partial\xi}\right)\frac{\partial T}{\partial\eta} = \frac{1}{RePrJ^2}\left(g_{22}\frac{\partial^2 T}{\partial\xi^2} + g_{11}\frac{\partial^2 T}{\partial\eta^2}\right). \tag{2.18}$$

Boundary conditions for pressure will be transformed as

$$R = R_1 : \frac{\partial p}{\partial \eta} = \frac{g_{11}}{Re\, J^2}\left(\frac{\partial x}{\partial \eta}\frac{\partial^2 u}{\partial \eta^2} + \frac{\partial y}{\partial \eta}\frac{\partial^2 v}{\partial \eta^2}\right); \tag{2.19}$$

$$R = R_2 : \frac{\partial p}{\partial \eta} \cong \left(\frac{\partial p}{\partial n}\right)_{pot}. \tag{2.20}$$

In these equations the elements of the metric tensor will have the form

$$g_{11} = \left(\frac{\partial x}{\partial \xi}\right)^2 + \left(\frac{\partial y}{\partial \xi}\right)^2; \quad g_{22} = \left(\frac{\partial x}{\partial \eta}\right)^2 + \left(\frac{\partial y}{\partial \eta}\right)^2. \tag{2.21}$$

The choice of transformations (2.10)-(2.12) renders the off-diagonal elements of the metric tensor zero, i.e., $g_{12} = g_{21} = 0$. That is the reason why the mixed second derivatives are missing from the Laplacian terms in equations (2.14), (2.15), (2.17)-(2.19). The transformation also ensures that the coefficients of the first order derivatives in the Laplacian terms in the above equations are zero (Fletcher, [12]). Since the mapping is given by elementary functions, the metric parameters and coordinate derivatives can be computed from closed forms, hence numerical differentiation leading to numerical errors can be avoided.

The grid aspect ratio AR [12], i.e., the ratio of the two sides of an elementary rectangle on the physical plane (see Figure 1), will have the form

$$AR = \sqrt{\frac{g_{22}}{g_{11}}} = \frac{f_\eta}{g_\xi} = \frac{\xi_{\max} \log\left(R_2/R_1\right)}{2\pi\eta_{\max}}. \tag{2.22}$$

It can be seen from equation (2.22) that the grid aspect ratio is constant over the whole computational domain. By choosing the number of grid points in the ξ and η directions properly, this constant can be set to unity resulting in conformal transformation [12].

4. Numerical method applied and computational results

4.1. **Computational method.** The author developed a computer code which is applicable to the computation of flow around a fixed or oscillating cylinder, or a cylinder in orbital motion. This code calculates the velocity, pressure, and time histories of lift and drag coefficients. Several other quantities are calculated, including the vorticity distribution, stream function, the location of the front stagnation point, and the lower and upper separation points changing with time. The code has recently been extended to compute the heat transfer between a heated cylinder and the fluid flowing around it.

The transformed governing equations are solved by the finite difference method. The time derivatives in the transformed Navier-Stokes equations (2.14), (2.15) and in energy equation (2.18) are approximated by forward differences. Fourth order central difference scheme is used for the diffusion terms and the pressure derivatives. The widely used modified third order upwind scheme proposed by Kawamura and

Kuwahara [4] proved to be successful in handling the convective terms in the Navier-Stokes and energy equations.

The equations of motion and equation of energy are integrated explicitly giving the velocity and temperature distributions at every time step. In the knowledge of the velocity distribution in an arbitrary time step, the pressure is calculated from equation (2.17) by using the successive over-relaxation method (SOR). Dilation D is chosen to be zero at every time step. The pressure on the cylinder surface is calculated by the third order formula at every time step, shear stress on the cylinder surface is derived from the velocity distribution, and from the pressure and the shear stress we can derive the time histories of lift and drag coefficients. The non-dimensional vortex shedding frequency, Strouhal number St can be determined from the location of the spectrum peak of FFT, when applied to the oscillating lift coefficient or other oscillating signals.

The dimensionless heat transfer coefficient or local Nusselt number Nu is obtained at every time step by using the temperature distribution

$$Nu\left(\varphi,t\right)=\frac{hd}{k}=-\left(\frac{\partial T}{\partial R}\right)_{w}=-\left(\frac{\partial T}{\partial \eta}\right)_{\eta=0}\left(\frac{d\eta}{dR}\right)_{\eta=0}=-\frac{1}{R_1 f'\left(\eta=0\right)}\left(\frac{\partial T}{\partial \eta}\right)_{\eta=0} \tag{2.23}$$

where $f'\left(\eta\right)$ means the first derivative of f with respect to η, and φ is the polar angle measured along the periphery of the cylinder. The average Nusselt number $Nu\left(t\right)$ is obtained as

$$Nu\left(t\right)=\frac{1}{2\pi}\int_{0}^{2\pi}Nu\left(\varphi,t\right)d\varphi.$$

The time averaged Nusselt number $\overline{Nu}$ can also be obtained by taking the average of $Nu\left(t\right)$ over a time period taken after reaching the quasi-steady state and covering more than one cycle.

The computational grid used is 241x131 O-mesh. The diameter of the outer boundary of computation is $30d$, where $d=2R_1$ is the cylinder diameter. Non-dimensional time steps used were usually 0.001 and 0.0005.

4.2. **Momentum transfer.** Previous studies by the author have focused upon the computation of momentum transfer between a uniform fluid flow and a single circular cylinder, either fixed or oscillating. Good agreement was obtained for fixed cylinders against experimental results for Strouhal number and mean drag coefficient versus Reynolds number [9]. Computations for oscillating cylinders were carried out to determine the amplitude threshold curve for locked-in vortex shedding due to crossflow cylinder oscillation [9].

A very important feature of momentum transfer is the base pressure coefficient. In Figure 2 time-mean values of the base pressure coefficients $\overline{C}_{pb}$, or the non-dimensional pressure measured at the farthest downstream point of the cylinder are compared with the experimental results of Roshko [10]. Agreement was good, but note that a discrepancy emerges at Re=180. This is fairly consistent with the findings of Williamson

[13] that flow becomes unstable and 3-D effects begin to appear above Re=160. This instability was accurately predicted by the computational code. Also shown in Figure 2 are the computed rms values of the base pressure C_{pbrms}. Below Re=60, this value is very small, which is in agreement with experimental findings that vortex shedding begins between Re=40 and 50 [3] or, as mentioned earlier, the value of 45.9 given in [7].

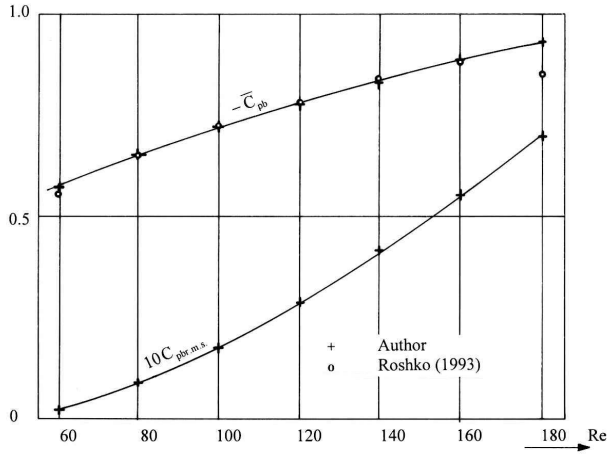


Figure 2. Comparison of base pressures

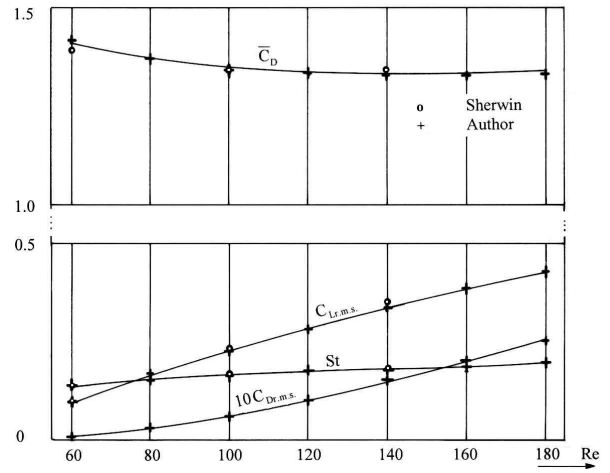


Figure 3. Computational results for fixed cylinder with comparisons

To further determine momentum characteristics of flow, computations of the time-mean value and root-mean-square (rms) value of drag coefficients ($\overline{C}_D$and C_{Drms}) and rms value of lift coefficient (C_{Lrms}) were made for a fixed cylinder at different Reynolds numbers. These and the non-dimensional vortex shedding frequency St

are shown in Figure 3, with comparisons with the unpublished computational results of S. Sherwin (Imperial College of Science, Technology and Medicine, using spectral element method) for Re=60, 100, and 140. As can be seen, the agreement is quite good.

4.3. **Heat transfer.** The cylinder surface is given a constant temperature, and the temperature of the fluid flowing around it is below the cylinder temperature, meaning that heat is transferred from the cylinder surface in the fluid. To keep a constant temperature, the cylinder is heated (as is the case with hot wire anemometer). Here, it is assumed that the temperature difference is not large enough to influence the properties of the fluid, and a constant property fluid was assumed in this study.

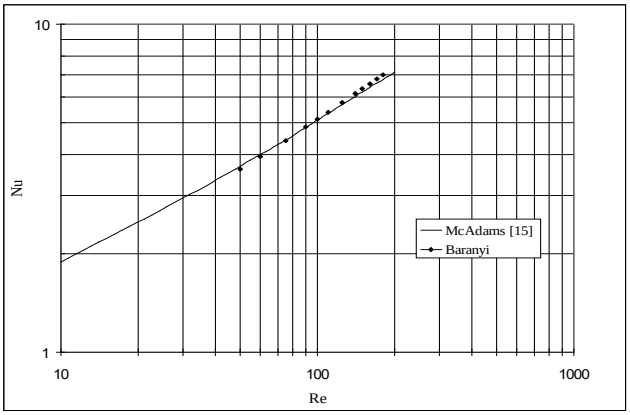


Figure 4. Measured and calculated time-averaged Nusselt number versus Reynolds number

Computations were carried out for fixed cylinders from Re=50 to 180, bearing in mind the fact that three-dimensional instability begins to occur at around Re=160, [13]. Figure 4 shows the author's computational results for time-averaged Nusselt numbers for different Reynolds numbers up to Re=180, and experimentally obtained time-averaged Nusselt number $\overline{Nu}$ for heating of air flowing across a single circular cylinder, versus Reynolds number, from Özisik [14] and McAdams [15]. As can be seen, the agreement is quite good, and possibly better than it appears in Figure 4. This is because the solid line is only a small part of a regression curve placed on experimentally measured values over an extremely wide Reynolds number domain, and the real measured values in the domain under consideration are above the regression curve itself, as are the computational results.

Computational results for which heat transfer has also been calculated are summarized in Table 1, as a function of Reynolds number. Strouhal number, time-mean values and rms values of the lift and drag coefficients, base pressure coefficient, and Nusselt number are given, with the exception of the time-mean value of the lift coefficient, which is zero. Note that values starting at Re=160 may be influenced by three-dimensional instability. It was found that Strouhal number derived from the

spectrum of the lift coefficient and that from the spectra of any other oscillating signal (C_D, C_{pb}, Nu) were somewhat different for Re=160 and above, and the averaged values for St are given in Table 1. This code predicts that the three-dimensional instability begins around Re=160.

Table 1. Effect of Re on momentum and heat transfer

Re	St	$\overline{C}_D$	$-\overline{C}_{pb}$	$\overline{Nu}$	C_{Lrms}	C_{Drms}	C_{pbrms}	Nu_{rms}
50	0.125	1.451	0.518	3.609	0.032	0.0001	0.0009	0.00004
60	0.137	1.419	0.574	3.950	0.093	0.0007	0.0022	0.0003
75	0.152	1.381	0.636	4.421	0.152	0.0024	0.0069	0.0008
90	0.163	1.357	0.687	4.856	0.199	0.0046	0.0129	0.0012
100	0.163	1.346	0.718	5.132	0.228	0.0064	0.0174	0.0014
110	0.171	1.338	0.747	5.396	0.256	0.0084	0.0226	0.0018
125	0.176	1.331	0.789	5.776	0.297	0.0117	0.0315	0.0027
140	0.179	1.325	0.830	6.136	0.337	0.0153	0.0413	0.0041
150	0.185	1.329	0.857	6.367	0.363	0.0178	0.0481	0.0056
160	0.188	1.330	0.883	6.590	0.388	0.0203	0.0552	0.0074
170	0.192	1.332	0.908	6.807	0.412	0.0229	0.0623	0.0094
180	0.195	1.334	0.933	7.018	0.435	0.0256	0.0696	0.0118

The rms values of the Nusselt number are shown in Figure 5. As can be seen, the amplitude of oscillation with time in the Nusselt number increases pronouncedly with increasing Reynolds number. This means that vortex shedding has an increased effect on heat transfer at larger Reynolds numbers.

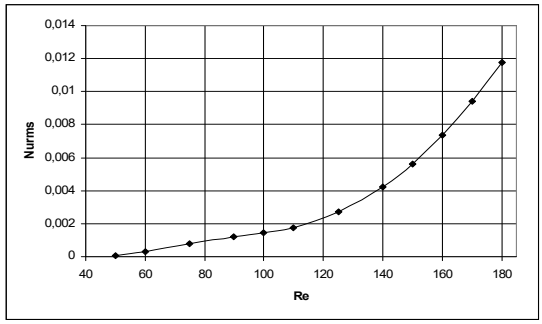


Figure 5. Root-mean-square value of Nusselt number versus Reynolds number

The time-history of the average Nusselt number Nu is shown for Re=150 in Figure 6. In the case of a fixed cylinder, the establishment of a quasi-steady state of vortex shedding takes a relatively long time. The frequency of oscillation in the Nusselt number is the same as the frequency of oscillation for the drag coefficient, while the frequency of oscillation of the lift coefficient is half that of Nu and the drag coefficient. This can be seen by taking the fast Fourier transform (FFT) of these three signals. Thus, in the case of the lift coefficient, one cycle consists of two vortices shed (clockwise

on the upper and counter-clockwise on the lower side), while for the drag coefficient and heat transfer, the shedding of a single vortex constitutes one cycle.

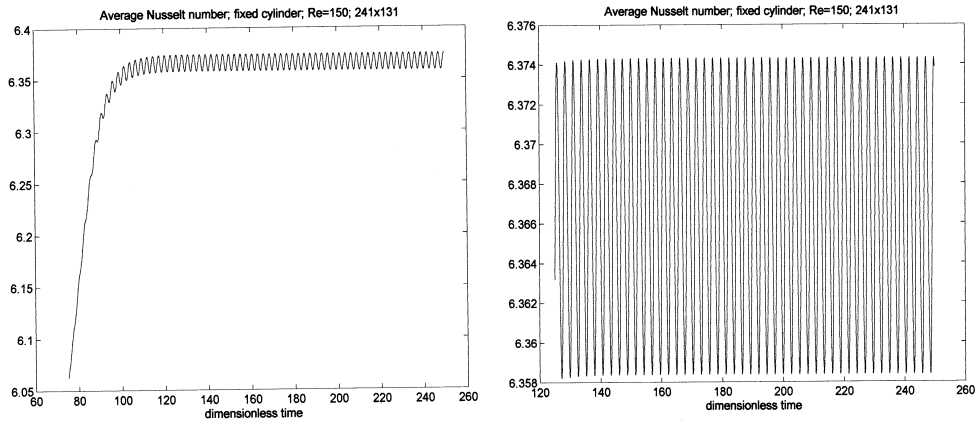
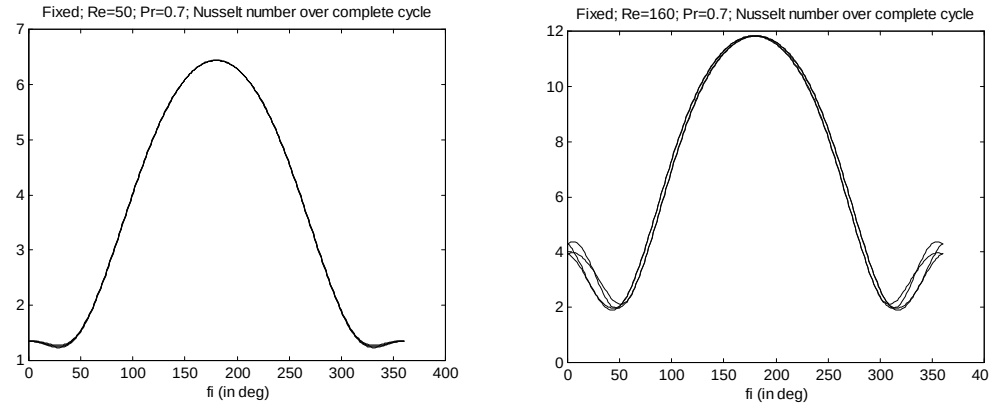


Figure 6. Time history of average Nusselt number



a) *Re*=50 b) *Re*=160

Figures 7a and b. Local Nusselt number

Figures 7a and 7b show the local Nusselt number distribution over the cylinder surface for a complete cycle of vortex shedding at dimensionless times t_0; $t_0 - \frac{T}{4}$; $t_0 - \frac{T}{2}$; $t_0 - \frac{3T}{4}$ where $t_0 = 250$ and T is the period of a shedding cycle. As can be seen, the curves are similar in shape and magnitude, but shift slightly over the whole periphery, and the shift is largest on the downstream side of the cylinder. At Re=50 the four curves belonging to the different phases almost completely coincide (Figure 7a), but the shift among the curves grows pronouncedly larger as the Reynolds number increases (see Figure 7b for an example). The maximum heat transfer rate is around $\varphi = 180^o$, near the upstream stagnation point. This can be attributed to the

thin boundary layer present at that point. As the boundary layer thickens, the local Nusselt number decreases steeply. The function seems to be completely symmetrical around this point, and minimum heat transfer occurs not at the exactly opposite downstream point, but around 45^o to each side of it. There is a local maximum in the heat transfer at the downstream point ($\varphi = 0^o$), probably due to periodic vortex shedding.

The curves in Figures 7a and 7b are very similar to those based on computational results obtained using the stream function-vorticity method [8]. These researchers showed results for Re=200, a somewhat unfortunate choice as they used a two-dimensional code.

5. **Concluding remarks**

The finite difference method was applied for the numerical simulation of unsteady laminar flow and forced convection from a fixed cylinder placed in a uniform flow. Primitive variable formulation was used for the fluid flow, and the fluid is assumed to be incompressible and of constant property. The viscous energy dissipation term was neglected in the energy equation since its value is small at low Reynolds numbers. By using boundary-fitted coordinates, interpolation of the boundary conditions becomes unnecessary. An orthogonal transformation provides a fine grid scale in the vicinity of the cylinder and a coarse grid in the far field. Time derivatives are approximated by forward differences, space derivatives by fourth order central differences, except for convective terms which were approximated by a third order modified upwind scheme.

The code developed was applied to the investigation of both flow around a circular cylinder and forced convection from the cylinder. The non-dimensional vortex shedding frequency (Strouhal number), time-mean values of drag and of base pressure coefficients, further the root-mean- square values of lift, drag, base pressure, and Nusselt number were determined for Reynolds numbers from 50 to 180. Where possible, results were compared with experimental data and excellent agreement was obtained, except for the vicinity of Re=180, where three-dimensional instability might have had an influence.

The distribution of the local Nusselt number over the cylinder surface was also investigated over a complete cycle. It was found that the curves belonging to different phases are similar in shape and magnitude, but shift slightly over the whole periphery of the cylinder, and the shift is largest on the downstream side of the cylinder. This shift increases with increasing Reynolds number.

The good agreement found between experimental and computational values encourages the author to extend the investigation in the future to the cases of forced convection from an oscillating cylinder, and to the three-dimensional case. Another future plan is to take into account the effect of temperature on the properties of the fluid.

Acknowledgement. The author would like to thank Dr. Spencer Sherwin for making the computations here, and for permitting him to use them. The support provided by the Hungarian National Research Foundation (Project No. T042961) is gratefully acknowledged.

References

1. Roshko, A.: *On the development of turbulent wakes from vortex streets*, NACA Rep., 1191, 1954.
2. Norberg, C.: *Flow around a circular cylinder: aspects of fluctuating lift,* Journal of Fluids and Structures, **15**, (2001), 459-469.
3. Bearman, P.W.: *Developments in the understanding of bluff body flows,* JSME Centennial Grand Congress, International Conference on Fluid Engineering, **1**, Tokyo, (1997), 53-61.
4. Kawamura, T. and Kuwahara, K. : *Computation of high Reynolds number flow around a circular cylinder with surface roughness*, Proceedings of the 22nd Aerospace Sciences Meeting, Reno, Nevada, AIAA-84-0340, (1984), 1-11.
5. Braza, M., Chassaing, P. and Minh, H. H.: *Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder*, Journal of Fluid Mechanics, **165**, (1986), 79-130.
6. Karniadakis, G.E. and Triantafyllou, D.S.: *Frequency selection and asymptotic states in laminar wakes.* Journal of Fluid Mechanics, **199**, (1989), 441-469.
7. Lange, C.F., Durst, F. and Breuer, M.: *Momentum and heat transfer from cylinders in laminar crossflow at* $10^{-4} \leq Re \leq 200$*,* International Journal of Heat and Mass Transfer, **41**, (1998), 3409-3430.
8. Mahfouz, F.M. and Badr, H.M.: *Forced convection from a rotationally oscillating cylinder placed in a uniform stream,* International Journal of Heat and Mass Transfer, **43**, (2000), 3093-3104.
9. Baranyi, L. and Shirakashi, M.: *Numerical solution for laminar unsteady flow about fixed and oscillating cylinders*, Computer Assisted Mechanics and Engineering Sciences, **6**, (1999), 263-277.
10. Roshko, A.: *Perspective on bluff body aerodynamics*, Journal of Wind Engineering and Industrial Aerodynamics, **49**, (1993), 79-100.
11. Harlow, F.H. and Welch, J.E.: *Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface*, Physics of Fluids, **8**, (1965), 2182-2189.
12. Fletcher, C.A.J.: *Computational Techniques for Fluid Dynamics*, Vol. 2, Springer, 2nd Ed., Berlin, 1997.
13. Williamson, C.H.K.: *Vortex dynamics in the cylinder wake*, Annual Review of Fluid Mechanics, **28**, (1996), 477-539.
14. Özisik, M.N.: *Heat Transfer, A Basic Approach,* McGraw-Hill, New York, 1985.
15. McAdams, W.H.: *Heat Transmission,* McGraw-Hill, New York, Third edition, 1954.

Journal of Computational and Applied Mechanics, Vol. 4., No. 1., (2003), pp. 27–36

OPTIMIZATION TECHNIQUE IN CASE OF TIME-DEPENDENT LOADING

Anna Vásárhelyi, János Lógó
Faculty of Civil Engineering, Budapest University of Technology and Economics
Műegyetem rkp. 3-5, 1521 Budapest, Hungary
vasarhelyi@epito.bme.hu, logo@ep-mech.me.bme.hu

[Received: June 14, 2002]

Abstract. The structural design problem is solved in case of time-dependent loading. The state variables are given in function space, where the Fritz-John theorem is not proved. Approximating the functions by generalized Fourier-series, the coefficients become unknowns. Transforming the problem into Banach space, where the Fritz-John theorem is proved, the problem is solvable.

Mathematical Subject Classification: 74C99,74P99
Keywords: Banach space, mathematical programming, time-dependent loading

1. Introduction

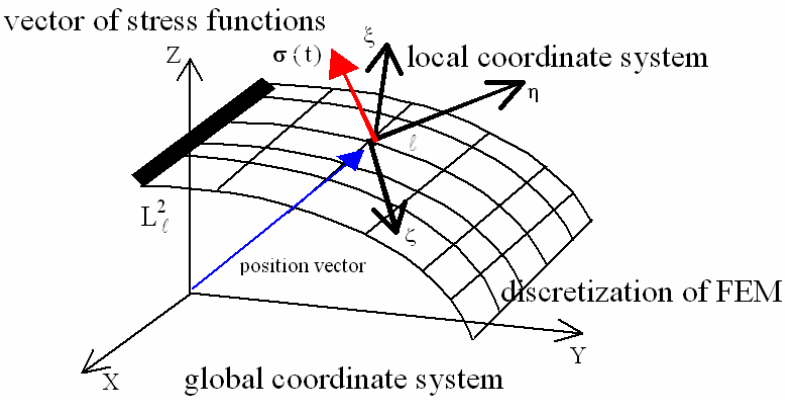
The aim of this research is to present some computational models of mechanical processes in case of structures and, in addition, the energy dissipation effects are taken into consideration. These models are used for designing the flexibility of the structure. The reason of this investigation is the fact that it is possible to determine only the relative weak minimum by means of the finite element method and the variation calculus [7], but in spite of this fact different variations of finite element methods are frequently used in the literature and practice [3-5,9,11]. In this paper elasto-plastic material models are used and the investigated computational models can be applied if the relative strong minimum of variation calculus is determined in case of analysis of structures. From these tendencies it follows that one should try to solve the problem with different types of approximations and numerical methods and to extend the finite element methods to determine a relative, strong or absolute minimum in spite of the fact that the results of these methods are valid in some cases – not to taking into consideration the approximations. Due to these reasons it is necessary to apply mechanical processes in the structural analysis with the aid of a mathematical background, which does not contain the problem mentioned above. The method of this research is based on partially the theoretical thermodynamics in physics concerning the mechanical processes [10,13-15]. It was necessary to look for a mathematical field, which patterns to the physical background and to elaborate the

mathematical branch, which is conformable for the numerical solution of the analysis of the mechanical processes by computer as well.

2. Mathematical background

The problem is written in function spaces, as a primal-dual form. It creates difficulties that the extreme theorems of mathematical programming have not been proved in function spaces.

The state variables (e.g. stresses, strains, etc.) are given in vector space by vector-scalar functions in case of equilibrium state. For a discretized structure this vector space is supposed to be an n-dimensional space in a global coordinate system denoted by X, Y, Z. Each node of the element is defined by a position vector (see Figure 1). A state vector is attached to each position vector in the local coordinate system ξ, η, ζ. The number of the independent components depends on the freedom of the nodes (6 for the example in Figure 1) and it equals n times the dimension of the state vector. In case of time-dependent problems, the state functions are given in both local and global coordinate systems and are vector-vector functions which depend on time. Within the context of small displacements theory, the position vectors are time independent.



The state variables are time-functions

The state variables are approximated by generalized Fourier-series. The complementary problem is transformed from the functions space into the Banach space of the Fourier coefficients, where the Fritz-John theorem has been proved. From the Banach space it is easy to switch to the infinite Euclidean space. The latter can be truncated to a finite space in which the problem becomes solvable [2].

Let us see an arbitrary vector-scalar function by dimension s. All of the components of this function are elements of the space $L^2(\Omega)$, $\Omega[0,1]$ [12]. The space $L^2(\Omega)$ is a Hilbert-space, it is possible to choose an orthogonal basis $P_i(t)$, $(i = 1, ..., \infty)$ on the

interval $\Omega = [0,1]$ in polynomial form. An element of this Hilbert-space is:

$$x(t) = \sum_{i=1}^{\infty} \alpha_i P_i(t), \quad P_i(t) \in L^2(\Omega), \quad \alpha_i \in \mathbb{R}, \quad i = 1, \ldots, \infty, \quad t \in [0,1], \tag{1}$$

where the set of real numbers is denoted by $\mathbb{R}$. The s dimensional function space contains the state variables, which belong to a node. The Hilbert-spaces of the structure are defined as a direct multiplication of the s dimensional function spaces powered n

$$(L_1^2 \times L_2^2 \times ... \times L_j^2 \times ... \times L_s^2)^n \tag{2}$$

One element of the space is the projection of the state variables of the j^{th} freedom:

$$r^j(t) = \sum_{k=1}^{3} x_j^k(t)\mathbf{i}_k = \sum_{k=1}^{3} \left(\sum_{i=1}^{\infty} \alpha_{ij} P_i^j(t) \right) \mathbf{i}_k \tag{3}$$

$$\alpha_{ij} \in \mathbb{R}, \quad P_i^j(t) \in L_j^2([0,1]), \quad t \in [0,1], \quad j = 1, ..., s, \quad i = 1, ..., \infty,$$

where $\mathbf{i}_k$ $(k = 1, ..., 3)$ are the unit vectors of the local coordinate-system, $x_j^k(t)$ is the function on the k^{th} axis of the local coordinate-system. $P_i^j(t)$ is the i^{th} element of the approximation polynomial system belonging to the k^{th} axis of the local coordinate system. α_{ij} denotes the i^{th} Fourier-coefficients of the polynomial $P_i^j(t)$ at the j^{th} freedom.

The Fourier coefficients are summarized into a matrix according to nodes:

$$[\tilde{\alpha}] = \begin{bmatrix} \alpha_{11}^1 & \ldots & \alpha_{\infty 1}^1 & \alpha_{12}^1 & \ldots & \alpha_{\infty 2}^1 & \alpha_{\infty s}^1 & \ldots & \alpha_{\infty s}^1 \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \alpha_{11}^n & \ldots & \alpha_{\infty 1}^n & \alpha_{12}^n & \ldots & \alpha_{\infty 2}^n & \alpha_{\infty s}^n & \ldots & \alpha_{\infty s}^n \end{bmatrix}. \tag{4}$$

The mechanical problem has been formed in the space $\mathbb{F}^\ell = (\mathbb{R}^3 \times L_1^2 \times L_2^2 \times ... \times L_j^2 \times ... \times L_s^2)^n$ and the computation has been done in the space of Fourier coefficients.

The extreme problem of the transformed Lagrangian function can be divided into primal and dual problems in mathematical programming. The primal variables are intensive ones, the dual variables are extensive ones, from a physical point of view. The objective function contains the sum of complementary strain energy and dissipative energy, respectively.

It has been proved, that the stationary curve characterized the process and the changing of state variables in time is solvable in linear and nonlinear cases using the Fritz-John condition by mathematical programming. The unknowns are the generalized Fourier coefficients of the Fourier series used for approximation. The inequalities can be taken into consideration in discretized time points only [16].

The Fritz-John condition can be written by Stieltjes' derivative, so the dual problem can be formed by the Wolfian procedure using Stieltjes' derivative. In this way the deductions are easier and clearer concerning the mechanical processes than to work in the space of the generalized Fourier coefficients.

Mechanically the primal problem is expressed by the equilibrium equations with boundary conditions as constraints, and by the function of the complementary strain

energy as objective function, in function space. The Wolfian procedure (for dual creation) results in the compatibility equations with the boundary conditions as constraints, and in the energy theorem of the elasticity as the objective function in function space.

The basic idea of the deductions are:

- the primal problem is based on extreme theorems of physics,
- using the Wolfian process the dual problem was formed and it was checked from a mechanical point of view.

The dual problem shows as a result that the analysis of a structure made of nonlinear elastic material with time dependent state variables has time-dependent energy function as the objective function, and the compatibility equations as constraints. The compatibility equations are equivalent to the Kuhn-Tucker conditions [6,8].

3. Elasto-plastic models

The structure is divided into finite elements in space. The state variables, ordered to the FEM nodes, are time-functions (Figure 1).

The use of the elaborated mathematical tools has been presented in different mechanical problems [15-18]. The dissipated energies are taken into consideration by internal variables, which contains the changing of the micro-level material structure into a macro level one.

The limitation of the presented model are:

1. at least one continuous component has to be assumed,
2. the small displacement theory is valid,
3. stability problems are neglected.

The aim is to design the flexibilities of the members of a structure in case of elasto-plastic material. The following data are given:

– the external load time-function,
– the geometrical data (transfer matrix between the nodes, boundary conditions),
– plastic yield limit,
– dissipate yield limit,
– limit of flexibility $[\hat{\mathbf{F}}]$,
– dissipate matrix $[\mathbf{A}]$, which contains dissipate properties of material.

The primal problem can be formulated as follows:

$$[\mathbf{B}]^* \sigma(t) \langle \rho \rangle + [\mathbf{B}]^* r(t) + \mathbf{p}(t) = 0\,, \tag{5a}$$

$$f_i(\sigma_k(t), k = 1, ..., z) \leqslant 0, \qquad i = 1, ..., G, \tag{5b}$$

$$\varphi_i(r_k(t), k = 1, ..., z) \leqslant 0 \qquad i = 1, ..., G, \tag{5c}$$

$$f_i(\sigma_k(t), k = 1, ..., z) r_{i_k}(t) = 0, i = 1, ..., G,\ k = 1, ..., z, \tag{5d}$$

$$[\mathbf{F}] - \left[\hat{\mathbf{F}}\right] \leqslant 0, \tag{5e}$$

$$\frac{1}{2}\sigma(t)^* [\mathbf{F}] \sigma(t) \langle\rho\rangle + \frac{1}{2} r(t)^* [\mathbf{A}] r(t) \to \min ., \ \forall t,\ t \in [t_1, t_2] \tag{5f}$$

The unknowns involve:

the vector of stress functions $\sigma(t)$,

the vector of dissipate force functions $r(t)$,

and the flexibility matrix $[\mathbf{F}]$,

where n is the number of the nodes, s is the freedom of nodal displacements, G is the total number of the Gaussian points defined on the elements, z is the freedom of the stresses defined in the Gaussian points, $[\mathbf{B}]^*$ is the transfer matrix of the structure (the dimension is $ns \times Gz$), $r(t)$ is the force type internal variable (the dimension is ns), $< \rho >$ is the diagonal matrix of the Gaussian weights (the dimension is $Gz, \times Gz$), σ(t) is the vector of the stresses (the dimension is Gz), $\mathbf{p}(t)$ is the vector of the external loads acting on the nodes (dimension is n.s), $[\mathbf{F}]$ is the flexibility matrix (the dimension is $Gz, \times Gz$), $[\hat{\mathbf{F}}]$ is the prescribed value of the flexibility matrix (the dimension is $Gz, \times Gz$), $[\mathbf{A}]$ is the matrix of the dissipate property of the phenomena (the dimension is $Gz, \times Gz$).

The mechanical interpretations of the problem (5a-g) are: (5a) is the equilibrium equation concerning time-functions of stresses and force type internal variables and external forces in the extended space (the number of the equation is ns). The boundary conditions (5b) are expressed by equality and/or inequality constraints. The plastic yield conditions in the Gaussian points of the structure are the inequalities (5b). The inequalities (5c) yield the size of the force type internal variables in the Gaussian points. The equalities (5d) take a switch role. (5e) contains the limit value of the flexibility. The objective function (5f) is the sum of the complementary strain energy and the complementary dissipation energy.

Let's see how the model (5a-f) works. A point is in the elastic state, if the plastic yield condition is an inequality in the point. The values of the force-type internal variables are zero owing to the switch conditions (5d). In the equality equations the equal sign is valid in the space F^n. If the plastic yield condition becomes an equality in a given point the force type internal variable can appear in equations (5a). This comes from equations (5d) for every Gaussian point and every stress freedom. The force type internal variables are not zero, that is equations (5a) become inequalities in the space F^n (there are no force type internal variables among the usual state variables in mechanics). This expresses the fact that a point in plastic state is in a non-equilibrium state. If the constraint (5c) is an equality the energy dissipation capacity of the material reaches its maximum in a given Gaussian point. If the constraint (5c) is an inequality, the given Gaussian point of the structure is either in the elastic state or in the plastic state and the material is able to dissipate.

The matrix $[\mathbf{A}]$ contains material constants, which characterize the energy dissipation ability of the phenomenon. Handling this type of energy is rather complicated because of the lack of the material constants.

Wolfe's dual problem of (5a-f) is formed as follows (the dual variables $\mathbf{u}(t)$, $\lambda(t)$, $\Psi(t)$ and $\mathbf{x}(t)$ belong to equality (5a), inequalities (5b), (5c) and equalities (5d), respectively, and the dual vectors have function elements) (5):

$$[\mathbf{B}]\, u(t) + [\mathbf{F}]\, \sigma(t)\, \langle \rho \rangle + (\lambda\ (t)^* + x(t)^* r(t\ ))\, \frac{\partial f(\sigma(t))}{\partial \sigma(t)} = 0, \tag{6a}$$

$$[\mathbf{B}]\, u(t) + \psi(t)^* \frac{\partial \varphi(r(t))}{\partial r(t)} + x(t)^* f(\sigma(t)) + r(t)^*\, [\mathbf{A}] = 0, \tag{6b}$$

Boundary conditions: (6c)

$$\lambda(t) \geqslant 0, \quad \psi(t) \geqslant 0, \tag{6d}$$

$$\lambda(t)_i f_i(\sigma_k(t), k = 1, ..., z) = 0, i = 1, ..., G, \tag{6e}$$

$$\psi(t)_i \varphi_i(r_k(t), k = 1, ..., z) = 0, i = 1, ..., G, \tag{6f}$$

$$[\mathbf{F}] - \left[\hat{\mathbf{F}}\right] \leqslant 0, \tag{6g}$$

$$-\frac{1}{2}\, \langle \rho \rangle\, \sigma(t)^*\, [\mathbf{F}]\, \sigma(t) - \frac{1}{2} r(t)^*\, [\mathbf{A}]\, r(t) + u(t)^* p(t) \to \min \tag{6h}$$

The objective function expresses the principle of energy conservation in the problem.

Unknowns:

- $u(t)$ vector of the displacement functions,
- $r(t)$ vector of dissipate force functions,
- $[\mathbf{F}]$ the flexibility,
- $\boldsymbol{\lambda}(t)$ plastic flow,
- $\boldsymbol{\psi}(t)$ dissipate flow

The mechanical interpretations of the problem (5a-h) are: (6a) expresses the compatibility between the displacements and the strains (the number of the equations is Gz). Equations (6b) express the compatibility due to the dissipation (the number of the equations is Gz). The boundary conditions are saved in original form (6.c). Inequalities (6d) give the sign constrain of the plastic and the dissipation multiplier, respectively. They show the direction of the process. (6e) and (6f) are the normality roles, (6g) gives a limit value for the flexibility, (6h) is the objective function.

Experiments are demanded to determine the coefficients of the material models. The experimental work has not belonged to this work.

4. **Numerical example**

As an illustration of the proposed computational methods, consider a three-supported beam with time-dependent loading. The data of the structure can be seen in Figure 2. The loads act on the nodes. The functions of the external loads are given by

$$P(t) = 7.650000 + 22.581250t + 18.300000t^2 - 1.09375t^3.$$

The material constant of the dissipation is considered by a diagonal matrix $[\mathbf{A}]$ and $\langle a_i \rangle = \langle 0.09 \rangle$.

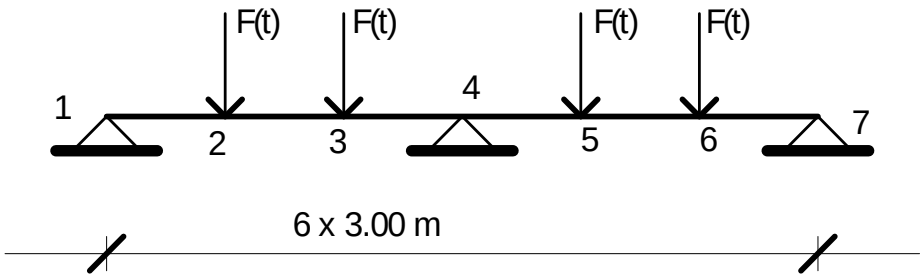


Figure 2. Test problem

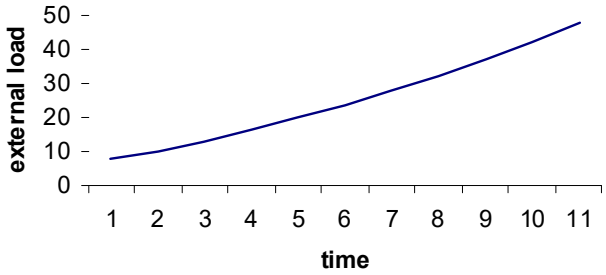


Figure 3. Load evaluation with respect to time

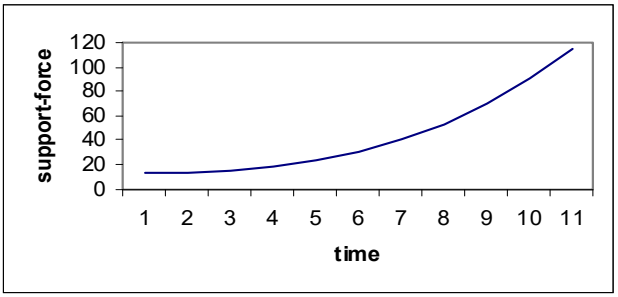


Figure 4. Variation of the reaction force at node 4

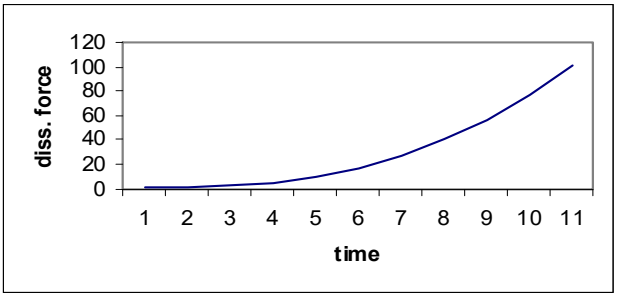


Figure 5. Variation of the dissipative force (internal variable) at node 4

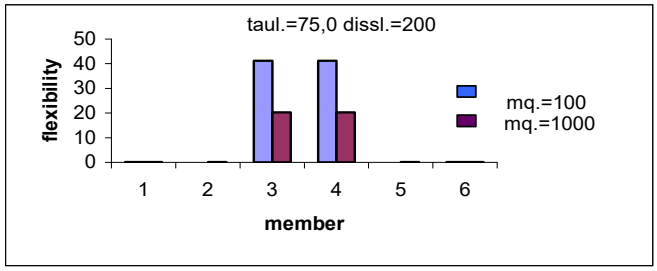


Figure 6. Variation of member flexibilities with different weight

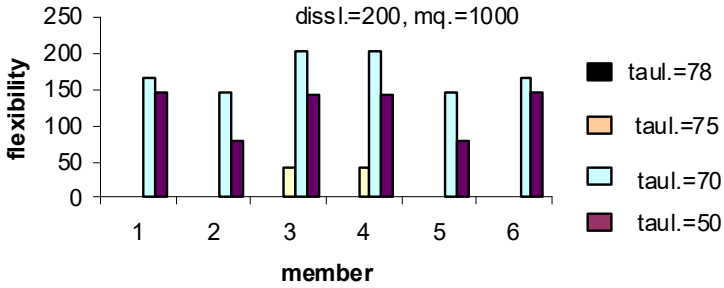


Figure 7. Variation of member flexibilities with yield limit

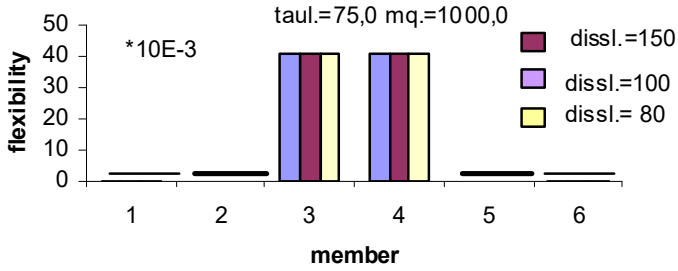


Figure 8. Variation of member flexibilities with different dissipative limit

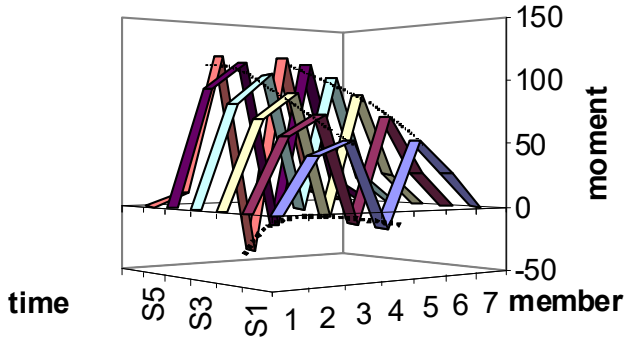


Figure 9. Variation of bending moments

The loads are approximated by the Legendre polynomial system up to four members. The variation of the external load function can be seen in Figure 3. The structure is divided into by seven nodes into 6 members. The unknowns are the moments and the shear forces at the Gaussian points of the members. On the basis of the general form, the statically admissible internal forces are determined by the equilibrium equations and force boundary conditions. The objective function contains the complementary potential energy. The displacement boundary conditions are taken into consideration in the objective function. In our example, the Huber-Mises-Hencky yield condition is applied. To solve this problem we used a nonlinear mathematical programming system [1]. The results can be seen in Figures 4-9. In Figures 4-5 the variations of the middle support reaction force and dissipative force (which is the internal variable) can be seen, respectively. The flexibilities of the members are described in Figures 6-8 with different yield and dissipative limits and the bending moments of the structure can be seen in Figure 9.

5. **Conclusion**

A flexibility design problem is solved in case of time-dependent loading. The function of the state variables is approximated by generalized Fourier series. Transforming the problem into Banach space, where the Fritz-John theorem is proved, the problem is solvable. The elastic and dissipative energies are taken into consideration. The theorem mentioned above and the algorithms of the nonlinear mathematical programming are used for the numerical solution. The numerical example demonstrates that the models are suitable for flexibility design.

Acknowledgement. The present study was supported by The Hungarian National Scientific and Research Foundation (OTKA) (Grant No. T029638) and the Hungarian Ministry of Education (F-33/00).

References

1. Bazarra, M.S. and Shetty, C.M.: *Nonlinear Programming, Theory and Algorithms*, John Wiley & Sons , New York, 1979.
2. Bhakta, P.C. and Roychandhuri, S.: *Optimization in Banach Spaces*, Journal of Math. Analysis and Applications, **134**, (1988), 460-470.
3. Cohn, M. Z. and Maier, G. (Eds.): *Engineering Plasticity by Mathematical programming*, Pergamon Press, New York, 1979.
4. De Donato, O.: *On Piecewise-Linear Constitutive Laws in Plasticity*, Tech. Rept., ISTC, 1974.
5. Kaliszky, S.: *Plasticity, Theory and Engineering Application*, Elsevier Science Publishers, Amsterdam, 1989.
6. Karush, W.: *Minima of Functions of Several Variables with Inequalities as Side Conditions*, M.S. Thesis, Dept. of Mathematics, University of Chicago, 1939.
7. Koiter, W.T.: *Stress-strain relations, uniqueness and variational theorems for elastic-plastic materials with singular yield surfaces*, Quart. Appl. Math., **11**(3), (1953), 350-354.

8. Kuhn, H.W. and Tucker, A.W.: *Nonlinear Programming*, 2nd Berkeley Symp. Math. Statistics and Probability, Berkeley, 1951.

9. Maier, G.: *A matrix structural theory of piecewise linear elastoplasticity with interacting yield planes*, Meccanica, **5**, (1970), 55-66.

10. Muschik, S.: *Non-equilibrium Thermodynamics with Applications to Solids*, Springer-Verlag, Wien, New York, 1993.

11. Sawczuk, A.: *Mechanics and Plasticity of Structures*, Ellis Horwood Ltd., Chichester, 1989.

12. Szőkefalvi-Nagy B.: *Real functions and Function Series*, Tankönyvkiadó, Budapest, 1961. (in Hungarian)

13. Vásárhelyi, A. and Lógó, J.: *Mathematical models of processes in mechanics*, Structural Optimization, **8**, (1994), 138-144.

14. Vásárhelyi, A. and Lógó, J.: *Elasto-Plastic Analysis of Structures Based on Thermodynamics by Mathematical Programming*, GAMM-ZAMM Wien, Proc. 69., 4-5, 1988.

15. Vásárhelyi, A. and Lógó, J.: *Time-dependent stress-limited mechanical modells of elasto-plastic processes*, Periodica Polytechnica Ser. Civil Eng., **38**(4), (1994), 431-444.

16. Vásárhelyi, A.: *A mathematical model for processes of structure analysis*, Computer Assisted Mechanics and Engineering Sciences, **3**, (1996), 297-316.

17. Vásárhelyi, A. and Lógó, J.: *The Effect of Energy Dissipation for the Analysis of Dynamically Loaded Elasto-Plastic Structures*, In the Second World Congress of Structural and Multidisciplinary Optimization, ed. by W. Gutkowski, Z. Mroz, IFTR, Wydawnictwo Ekoinzynieria Press, Warsaw, Poland, 405-410, 1997.

18. Vásárhelyi, A. and Lógó, J.: *Optimal Design of Structures Based on Energy Theorems in Case of Time-Dependent Loading with Dissipation*, 8^{th} AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization Long Beach, USA CD A00-400916-8 Sept. 2000.

Journal of Computational and Applied Mechanics, Vol. 4., No. 1., (2003), pp. 37–51

STABILITY ANALYSIS OF AN ORTHOTROPIC PLATE VIA MATHEMATICA

György Popper
Research Group for Computational Structural Mechanics, Hungarian Academy of Sciences
Budapest University of Technology and Economics
Műegyetem rkp. 3-5, 1111 BUDAPEST, Hungary
gypopper@epito.bme.hu

Béla Paláncz
Department of Photogrammetry and Geoinformatics
Budapest University of Technology and Economics
Műegyetem rkp. 3-5, 1111 BUDAPEST, Hungary
palancz@epito.bme.hu

Zsolt Gáspár
Department of Structural Mechanics, Budapest University of Technology and Economics
Műegyetem rkp. 3-5, 1111 BUDAPEST, Hungary
gaspar@ep-mech.me.bme.hu

[Received: June 16, 2002]

Abstract. Stability analysis of an orthotropic plate is studied using the algebra system *Mathematica*. The critical force is computed for given material parameters, geometry and load type (*direct problem*). Then the critical force is assumed to be known and the material and geometric parameters are computed (*inverse problem*). The inverse problem can be treated similarly to the direct problem because the numerical solution of both problems can be reduced to the symbolic-numerical solution of a matrix eigenvalue problem.

Mathematical Subject Classification: 35P05, 74K20, 74S99, 74G60
Keywords: orthotropic plate, structural stability, Mathematica system

1. Introduction

Bridge construction is one of the most important application fields of orthotropic plates used as roadway plates. Orthotrophy is achieved by transversal ribs resulting in different stiffnesses in the directions x and y, respectively. Employing the Huber equation [1] , Szabó [2] developed a numerical technique based on matrix algebra to study such plates a load in their plane. Later, it was realized that stability analysis is also necessary because of the heat dilatation phenomena (Popper [3]). In this paper we shall consider a slightly modified, more general version of the Huber equation, namely:

$$\partial_{x,x}\left(A\partial_{x,x}w\left[x,y\right]+2H\partial_{y,y}w\left[x,y\right]\right)+\partial_{y,y}\left(B\partial_{y,y}w\left[x,y\right]+g\left(x,P\right)w\left[x,y\right]\right)=0 \quad (1)$$

Herer w is the displacement function. A and B denote bending stiffnesses in the directions x, y, respectively, and H is the torsional stiffness. The function $g(x, P)$ represents the distribution of load on the edges of plate (see Figure 1). The (1) is associated with linear boundary conditions on the boundary S:

$$lw(x, y)|_S = 0\,. \tag{2}$$

Concerning the stability problem in this model one may consider P, the parameter of the edge force distribution, to be a critical parameter. This is the classical direct problem, namely to compute the critical load. The other possibility is to consider one of the parameters A, B i.e. the stiffnesses of the plate, which is subjected to a constant load, to be a critical parameter. This is the inverse stability problem.

In this paper we suggest a general method to handle the direct and inverse stability problems which are of the same form from a mathematical point of view. The method fully utilizes the symbolic and numerical computation capabilities of the integrated computer algebra system *Mathematica*.

2. Mathematical background

In the following some introductory concepts from Functional Analysis and the Galerkin method, which we shall apply, are presented shortly.

Let Ω be a constrained domain in the plane xy, and let $L_2(\Omega)$ denote the set of all square-integrable (real) functions defined over the domain Ω. If $u \in L_2(\Omega)$, then $\int\limits_\Omega u^2 d\Omega < \infty$.

The set $L_2(\Omega)$ is a vector space in which the *scalar product* is defined by the formula

$$(u|v) = \int\limits_\Omega uv\, d\Omega\,,$$

which induces a *norm* given by the formula

$$\|u\| = \sqrt{(u|u)} = \sqrt{\int\limits_\Omega u^2\, d\Omega}\,.$$

The vectors $u,\ v \in L_2$ are said to be *orthogonal*, if $(u \mid v) = 0$. Their *distance* is defined by the norm $||\, u - v\,||$. Any sequence of *linearly independent vectors* $\{\varphi_1, \varphi_2, \ldots\}$ in the function space $L_2(\Omega)$ form a *basis* for L_2 if every vector $u \in L_2$ can be approximated with arbitrary precision by a linear combination of functions $\{\varphi_1, \varphi_2, \ldots\}$. In other terms, if for every function $u \in L_2(\Omega)$ and arbitrary number $\varepsilon > 0$ there exists a positive integer N and scalar coefficients $c_1, c_2, \ldots, c_n$ such, that for every index $n > N$ it holds that

$$\left\| u - \sum_{k=1}^{n} c_k^{(n)} \varphi_k \right\| < \varepsilon.$$

The following *theorem* is a well-known result from the functional analysis:

If $v \in L_2$ is orthogonal to each element of a basis $\{\varphi_1, \varphi_2, \dots\}$ in L_2, then v is the null-element of the space L_2. In other terms if

$$(v|\varphi_k) = 0,\ k = 1, 2, ... \Rightarrow v = \Theta\,.$$

Suppose that in the function space $L_2(\Omega)$ is given the equation

$$Lu = f\,,$$

where L is a differential operator, which transforms the unknown function u into the given function f. The solution u is supposed to satisfy the prescribed boundary conditions on the boundary S of the domain Ω. If we find a function $u^* \in L_2$ which satisfies the (infinite) set of equations

$$(Lu^* - f|\varphi_k) = 0, \quad k = 1, 2, ...,$$

then it is a consequence of the previous theorem that $Lu^* - f = \Theta$. In other terms u^* is the solution of the equation $Lu = f$ in the space L_2. This simple idea forms the basis for the *Galerkin method*: The *approximate* solution of the equation $Lu = f$ is searched for in the form:

$$u_n = \sum_{k=1}^{n} c_k \varphi_k,$$

where n is an arbitrarily chosen, but fixed positive integer. In other terms, the approximate solution u_n of the exact solution $u^* \in L_2$ is expressed in the n-dimensional subspace of L_2 spanned by the base vectors $\varphi_1, \varphi_2, \dots, \varphi_n$. The unknown coefficients $c_1, c_2, \dots, c_n$ are computed from the orthogonality condition

$$(Lu_n - f|\varphi_k) = 0,\ k = 1, 2, ..., n,$$

which is a system of n equations. If the operator L is linear, then

$$Lu_n = \sum_{i=1}^{n} c_i L\varphi_i,$$

and hence this equation can be written in the usual form of a set of linear equations

$$\sum_{i=1}^{n} c_i\,(L\varphi_i|\varphi_k) = (f|\varphi_k)\,, \quad k = 1, 2, ..., n,$$

or in matrix form

$$\mathbf{Mc} = \mathbf{z},$$

where

$$\mathbf{M} = \begin{bmatrix} (L\varphi_1|\varphi_1) & \cdots & (L\varphi_n|\varphi_1) \\ \vdots & \ddots & \vdots \\ (L\varphi_1|\varphi_n) & \cdots & (L\varphi_n|\varphi_n) \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} (f|\varphi_1) \\ \vdots \\ (f|\varphi_n) \end{bmatrix}.$$

In our problem the operator L is given by (1). With respect to the relation

$$H = \kappa\sqrt{AB},$$

and taking into account that there is no load perpendicular to the surface of the plate, we have

$$f = 0\,.$$

Hence our problem is reduced to solving the set of homogeneous linear equations

$$\mathbf{Mc} = \mathbf{0}. \tag{3}$$

It is well known that a non-trivial solution exists if and only if the matrix $\mathbf{M}$ is singular. Consequently, the condition for the lose of stability is given by the equation

$$det[\mathbf{M}(A, B, P)] = 0.$$

If two variables from among A, B, P are known, then the unknown one can be determined from the previous equation, which is a polynomial equation of degree n in the unknown variable. Its smallest root is the critical value sought. The solution $\mathbf{c}$ of the corresponding set of equations $\mathbf{Mc} = \mathbf{0}$ can be normalized to be unique.

3. Application to an orthotropic plate

3.1. **A simple problem.** To demonstrate the suggested method, a simple problem was considered using notations of *Mathematica*. Figure 1 shows the geometry and boundary conditions as well as the type of load.

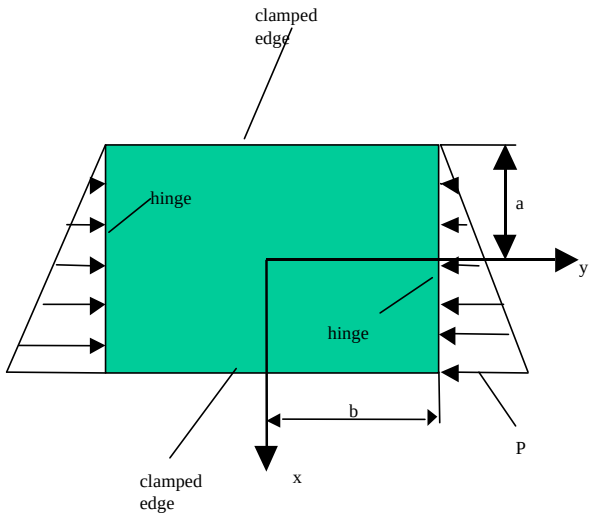


Figure 1. The geometry, boundary conditions and the type of load of the plate

The modified Huber equation is an elliptical PDF of order four associated with a linear force distribution on the edge

$$\partial_{x,x}\left(A\partial_{x,x}w\left[x,y\right]+2H\partial_{y,y}w\left[x,y\right]\right)+\partial_{y,y}\left(B\partial_{y,y}w\left[x,y\right]+\frac{P}{2}\left(x+a\right)w\left[x,y\right]\right)=0.$$

The homogeneous boundary conditions are the following:

– there is no displacement at the edges

$$w\left[a,y\right]=w\left[-a,y\right]=w\left[x,b\right]=w\left[x,-b\right]=0,$$

– edges parallel with the y axes are clamped

$$\partial_x w\left[a,y\right]=\partial_x w\left[-a,y\right]=0,$$

– there are hinges at the edges parallel with the x axes

$$\partial_{y,y}w\left[x,b\right]=\partial_{y,y}w\left[x,-b\right]=0.$$

The Galerkin method has been used to solve this homogeneous boundary value problem.

3.2. **Trial functions (basis).** Let us consider a polynomial of degree n in direction x and a polynomial of degree m in direction y, respectively, with dimensionless variables $\eta=x/a$ and $\xi=y/b$ (mind that the following formulas represent only one term).

$$\texttt{X[}\eta\texttt{\_,n\_]:=}\sum_{i=0}^{3}\alpha_i\eta^i+\eta^n$$
$$\texttt{Y[}\xi\texttt{\_,m\_]:=}\sum_{i=0}^{3}\beta_i\xi^i+\xi^m$$

The trial function of two variables is:

```
W[ξ_, η_,n_,m_]:=X[ξ,n]*Y[η,m]
W[ξ,η,4,5]
```
$$\left(\eta^4+\alpha_0+\eta\alpha_1+\eta^2\alpha_2+\eta^3\alpha_3\right)\left(\xi^5+\beta_0+\xi\beta_1+\xi^2\beta_2+\xi^3\beta_3\right)$$

The coefficients $\alpha[i]$ and $\beta[i]$ are defined by the eight boundary conditions.

```
eq1[n_]:=X[-1,n]=0
eq2[n_]:=X[1,n] =0
eq3[m_]:=Y[-1,m]=0
eq4[m_]:=Y[1,m] =0
eq5[n_]:=(D[X[η,n],η]/.η →-1)=0
eq6[n_]:=(D[X[η,n],η]/.η → 1)=0
eq7[m_]:=(D[Y[ξ,m],ξ,ξ]/.ξ →-1)=0
eq8[m_]:=(D[Y[ξ,m],ξ,ξ]/.ξ → 1)=0
```

Note, that the chain rule should be used because the dimensionless variables were introduced, but now, in case of homogeneous boundary conditions, it has no effect. Solving this set of equations, one obtains the unknown coefficients α[i] and β[i] depending on degrees n and m:

Let us consider $\max(n) = 5$ and $\max(m) = 5$:

```
maxn:=5;maxm=5;k=1;
Do[{α0,α1,α2,α3,β0,β1,β2,β3} =
 {α0,α1,α2,α3,β0,β1,β2,β3}/.solαβ[i,j]//Flatten;
 φ[k]=W[η,ξ,i,j];
 {α0,α1,α2,α3,β0,β1,β2,β3} = .;
 k=k+1,{i,4,maxn},{j,4,maxm}];
```

The number of the trial functions is:

```
K=(maxn-3)(maxm-3)
4
```

For example, the trial function φ[4]=W[η,ξ,5,5]:

```
φ[4]
```

$$\left(\eta - 2\eta^3 + \eta^5\right)\left(\frac{7\xi}{3} - \frac{10\xi^3}{3} + \xi^5\right)$$

```
Plot3D[φ[4],{η,-1,1},{ξ,-1,1}],PlotPoints→{30,30},
AxesLabel→{"η","ξ",None},
ColorFunction→(RGBColor[#,1-#,1-0.5#] & )];
```

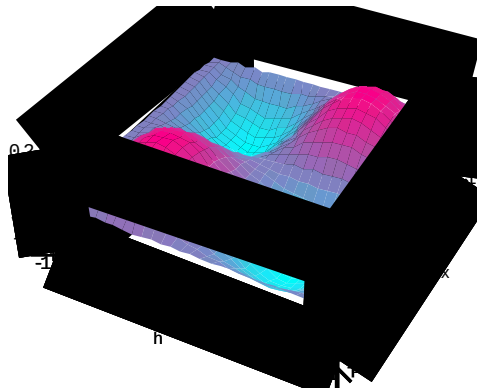


Figure 2. Trial function, n = 5, m = 5

4. **Computing system matrix**

A general element of the system matrix **M** is:

```
M[i_,j_]:  =
Simplify[
∫_-1^1 ∫_-1^1 (b^4 AD[φ[i],{η,4}]+2a^2 b^2 HD [φ[i],{η,2},{ξ,2}]+
Ba^4 D[φ[i],{ξ,4}] + Pa/2 (η+1)a^4 b^2 D[φ[i],{ξ,2}])
φ[j]dηdξ
```

Keep in mind that the chain rule should be used for $d\eta$ and $d\xi$, because the dimensionless variables were introduced, but now, in case of homogeneous equation (see (3)), it has no effect.

The matrix can be developed easily

```
sys=Table[M[i,j],{i,K},{j,K}];
```

The plate loses stability if the equation system has a nontrivial solution, namely its determinant is zero:

```
sysdet=Det[sys]
```

$$\frac{1}{5294745225}\left(268435456\left(70\mathrm{Ab}^4+220\mathrm{a}^4\mathrm{B}+132\mathrm{a}^2 2\mathrm{b}^2 2\mathrm{H}-11\mathrm{a}^5\mathrm{b}^2\mathrm{P}\right)\right.$$
$$\left(50\mathrm{Ab}^4+20\mathrm{a}^4\mathrm{B}+44\mathrm{a}^2\mathrm{b}^2\mathrm{H}-\mathrm{a}^5\mathrm{b}^2\mathrm{P}\right)$$
$$\left(\frac{1031865892864\mathrm{A}^2\mathrm{b}^8}{3472875}+\frac{4127463571456\mathrm{a}^4\mathrm{Ab}^4\mathrm{B}}{63669375}+\right.$$
$$\frac{4294967296\mathrm{a}^8\mathrm{B}^2}{3031875}+\frac{24897925414912\mathrm{a}^2\mathrm{Ab}^6\mathrm{H}}{121550625}$$
$$\frac{146028888064\mathrm{a}^6\mathrm{b}^2\mathrm{BH}}{9095625}+\frac{1241245548544\mathrm{a}^4\mathrm{b}^4\mathrm{H}^2}{40516875}$$
$$\frac{17541720178688\mathrm{a}^5\mathrm{Ab}^6\mathrm{P}}{1337056875}-\frac{36507222016\mathrm{a}^9\mathrm{b}^2\mathrm{BP}}{63669375}$$
$$\left.\frac{620622774272\mathrm{a}^7\mathrm{b}^4\mathrm{HP}}{191008125}+\frac{155155693568\mathrm{a}^{10}\mathrm{b}^4\mathrm{P}^2}{2941525125}\right)-$$
$$\frac{1}{72765}\left(16384\mathrm{a}^5\mathrm{b}^2\mathrm{P}\left(\frac{16906090788683776\mathrm{a}^5\mathrm{A}^2\mathrm{b}^{10}\mathrm{P}}{252703749375}+\right.\right.$$
$$\frac{67624363154735104\mathrm{a}^9\mathrm{Ab}^6\mathrm{BP}}{4632902071875}$$
$$\frac{70368744177664\mathrm{a}^{13}\mathrm{b}^2\mathrm{B}^2\mathrm{P}}{220614384375}+\frac{37084328181628928\mathrm{a}^7\mathrm{Ab}^8\mathrm{HP}}{804057384375}+$$
$$\frac{2392537302040576\mathrm{a}^{11}\mathrm{b}^4\mathrm{BHP}}{661843153125}+\frac{20336567067344896\mathrm{a}^9\mathrm{b}^6\mathrm{H}^2\mathrm{P}}{2948210409375}$$
$$\frac{287403543407624192\mathrm{a}^{10}\mathrm{Ab}^8\mathrm{P}^2}{97290943509375}-\frac{598134325510144\mathrm{a}^{14}\mathrm{b}^4\mathrm{BP}^2}{4632902071875}-$$
$$\left.\left.\frac{101682835336724484\mathrm{a}^{12}\mathrm{b}^6\mathrm{HP}^2}{138987706215625}+\frac{2542070883418112\mathrm{a}^{15}\mathrm{b}^6\mathrm{P}^3}{214040075720625}\right)\right)$$

5. **Direct stability problem**

For a direct stability problem, the geometrical parameters *a, b, A, B, H* are all known, and the load parameter P should be computed. Let us consider the following data taken from [3]:

```
κ=1.02848; a=5.7; b=11.4;
```

Then the system matrix $\mathbf{M}$ is

```
sysP=sys/.{A->190,B->230,H->κ√190230};
```

which can be decomposed as $\mathbf{sysP} = \mathbf{Z0} + P\mathbf{Z1}$, where

```
Z0 = Map[Coefficient[#,P,0] & ,sysP];Z0//MatrixForm
```

$$\begin{pmatrix} 2.3752\times10^{9} & 0 & 0 & 0 \\ 0 & 8.95871\times10^{7} & 0 & 0 \\ 0 & 0 & 1.57284\times10^{9} & 0 \\ 0 & 0 & 0 & 4.62149\times10^{7} \end{pmatrix}$$

```
Z1=Map[Coefficient[#,P] & ,sysP];Z1//MatrixForm
```

$$\begin{pmatrix} -1.97549\times10^{7} & 0 & -1.7959\times10^{6} & 0 \\ 0 & -1.93675\times10^{6} & 0 & -176068 \\ -1.7959\times10^{6} & 0 & -1.7959\times10^{6} & 0 \\ 0 & -176068 & 0 & -176068 \end{pmatrix}$$

Indeed

```
Simplify[sysP==Z0+PZ1]
True
```

Considering that **Z1** is not a diagonal matrix, the problem $\mathbf{Mc} = \mathbf{0}$ is a *generalized (linear)* eigenvalue problem for P as eigenvalue.

Then the critical load parameter is:

```
solP=Solve[Det[sysP]==0,P]
{{P → 45.3935}, {P → 118.547}, {P → 294.219}, {P → 977.085}}
Pcrit=Min[P/.solP]
45.3935
```

6. Inverse stability problem

In this case the critical load parameter is prespecified, and one of the geometrical parameters can be computed. For example, let us consider A_{crit} to be determined when

```
Pcrit = 55.;
```

Now, the system matrix, $\mathbf{M}$ is

```
sysA=sys/.{P->Pcrit,B->230,H->κ√A230};
```

which can be decomposed as $\mathbf{sysA} = \mathbf{S0} + \sqrt{A}\mathbf{S1} + A\mathbf{S2}$, where

```
S0=Map[Coefficient[#,A,0] & ,sysA];S0//MatrixForm
```

$$\begin{pmatrix} -1.05621\times10^{9} & 0 & -9.87744\times10^{7} & 0 \\ 0 & -9.44947\times10^{7} & 0 & -9.68376\times10^{6} \\ -9.87744\times10^{7} & 0 & -9.60192\times10^{7} & 0 \\ 0 & -9.68376\times10^{6} & 0 & -8.59042\times10^{6} \end{pmatrix}$$

```
S1=Map[Coefficient[#,√A ] & ,sysA];S1//MatrixForm
```

$$\begin{pmatrix} 1.9966 \times 10^7 & 0 & 0 & 0 \\ 0 & 1.95745 \times 10^6 & 0 & 0 \\ 0 & 0 & 6.65533 \times 10^6 & 0 \\ 0 & 0 & 0 & 652483. \end{pmatrix}$$

```
S2=Map[Coefficient[#,A] & ,sysA];S2//MatrixForm
```

$$\begin{pmatrix} 1.08931 \times 10^7 & 0 & 0 & 0 \\ 0 & 266204. & 0 & 0 \\ 0 & 0 & 7.78077 \times 10^6 & 0 \\ 0 & 0 & 0 & 190146. \end{pmatrix}$$

Indeed

```
Simplify[sysA==S0+√AS1+AS2]
True
```

Now, however **S1** and **S2** are diagonal matrices, the problem $\mathbf{Mc} = \mathbf{0}$ is a *nonlinear (quadratic) eigenvalue problem* for $\sqrt{A}$ as eigenvalue.

We can solve the equation, but now for A.

```
solA=Solve[Det[sysA]==0,A]
{{A → 8.4674} , {A → 22.6257} , {A → 81.8672} , {A → 246.723}}
```

There arise a problem to determine which solution is acceptable from engineering point of view. To solve this problem, let us consider the following figure:

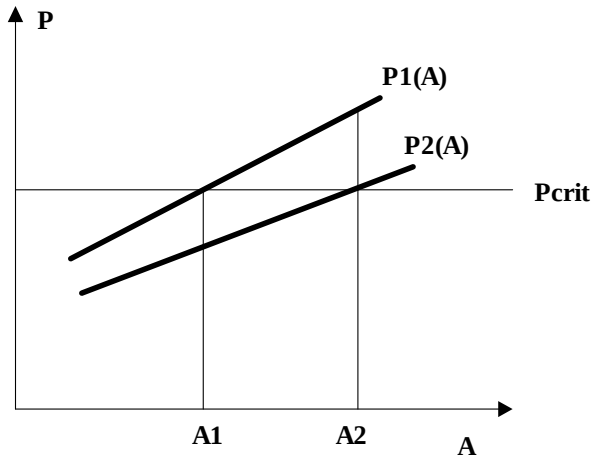


Figure 3. Selecting the proper critical geometrical parameter

It is easy to see that the index of the proper A_i can be computed as:

$$\min_j \left(P_j\left(A_i\right)\right) = P_{crit}.$$

Therefore, we can solve our stability equation symbolically for P with parameter A, namely $P_j(A)$:

```
solPA=Solve[sysdet==0,P]/.{B→230,H->κ√A230};
solPA=P/.solPA;
```

Then the matrix $m_{i,j} = P_j(A_i)$ can be computed as:

```
MPAcrit=Transpose[Map[Chop[N[#/.solA]] & ,solPA]]
```

$$\begin{array}{cccc} \{\{8.96183, & 9.78261, & 30.3149, & 55.0\}, \\ \{13.6388, & 18.5023, & 55.0, & 131.104\}, \\ \{26.0052, & 55.0, & 144.231, & 435.136\}, \\ \{55.0, & 151.326, & 370.553, & 1259.20\}\} \\ . & . & . & . \end{array}$$

Select the proper row:

```
rPAcrit=Select[MPAcrit,Chop[Min[#]-Pcrit=0 & ]
```

$\{\{55.,\ 151.326,\ 370.553,\ 1259.02\}\}$

Select the index of the row:

```
pAcrit=First[Position[MPAcrit,First[rPAcrit]]]
```

$\{4\}$

Select the proper critical geometrical parameter:

```
Acrit=A/.solA[[First[pAcrit]]]
```

246.723

We remark that **Z0** is a positive definite diagonal and **Z1** is a real symmetric matrix. Hence the eigenvalue problem can be reduced to a special eigenvalue problem for a symmetric, i.e., selfadjoint matrix.

7. Computing eigenshapes

Let us return to the direct stability problem. After the critical load is determined, one may compute the solution of (3), i.e., the values of c_i's. Unfortunately, in both cases (direct and inverse stability problem) we have a non-standard eigenvalue problem, therefore the usual techniques cannot be directly applied [5]. One may solve the problem by definition. One of the rows of **M** can be left out, and instead of it the normalization can be considered to be a new equation:

$$\sum_{i=1}^{K} c_i = 1\,.$$

However, one should carefully select the row to be left out, because of the special chess table structure of **M** (see matrix **Z**). One should leave out a proper row, which really cancels the singularity of **M**. Fortunately, *Mathematica* has a built-in function to compute the nullspace [6] of **M**, so one can easily solve this unpleasant problem:

```
c=NullSpace[sysP/.P→Pcrit][[1]]//N
```

$\{0., 4.78239, 0., 1.\}$

Norming:

```
< < LinearAlgebra'MatrixManipulation'
c / VectorNorm[c,2]
```

$\{0., 0.97883, 0., 0.204674\}$

Here we have to mention that in case of computation with non rational numbers, *NullSpace* may result in an empty set because of the ill-conditioned feature of $\mathbf{M}$. In that case the *Jordan decomposition* has been proved to be more robust and can provide acceptable approximation:

```
c = Transpose[
JordanDecomposition[(sysP//N)/.P→Pcrit][[1]]][[K]]
```

$\{0., -0.97883, 0., -0.204674\}$

Now we have the same result because we used rational numbers in the computation. In the following sections we employ Jordan decomposition in order to reduce computation time.

Then the first eigenshape:

```
wcrit = Σ_{i=1}^{K} c[[i]]φ[i]
```

$$0.(1-2\eta^2+\eta^4)(5-6\xi^2+\xi^4)+0.(\eta-2\eta^3+\eta^5)(5-6\xi^2+\xi^4)-$$
$$0.978783(1-2\eta^2+\eta^4)\left(\frac{7\xi}{3}-\frac{10\xi^3}{3}+\xi^5\right)-$$
$$0.204674(\eta-2\eta^3+\eta^5)\left(\frac{7\xi}{3}-\frac{10\xi^3}{3}+\xi^5\right)$$

```
Plot3D[wcrit,{η,-1,1},{ξ-1,1},PlotPoints->{30,30},
AxesLabel-> {"x/a","y/b",None},
ColorFunction->(RGBColor[#,1-#,0] & )];
```

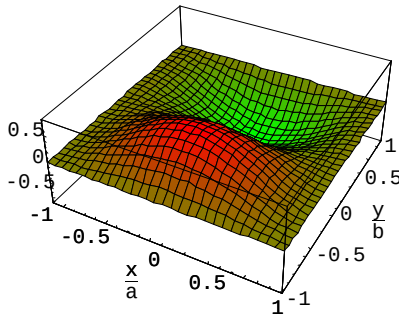


Figure 4. The first eigenshape

The contour plot shows clearly the effect antisymmetrical load distribution

```
ContourPlot[wcrit,{η,-1,1},{ξ-1,1},PlotPoints->50,
AxesLabel->{"a","b",None},
ColorFunction->(RGBColor[#,1-#,0] & ),Contours->15];
```

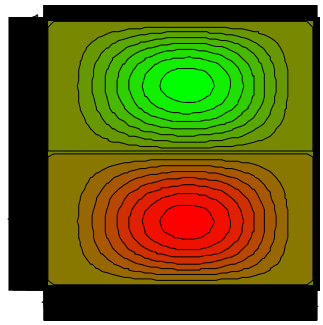


Figure 5. Contour plot of the first eigenshape

The further eigenshapes belonging to the next three roots are:

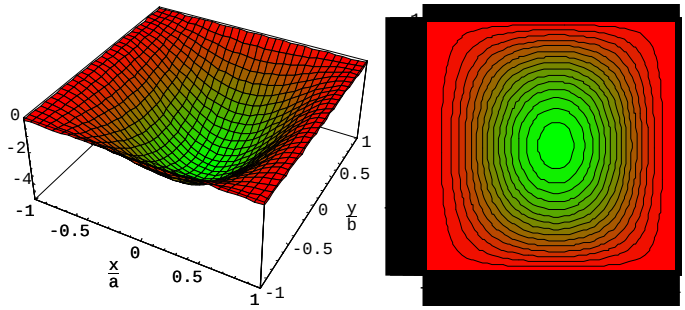


Figure 6a. Eigenshape for $P = 118.547$

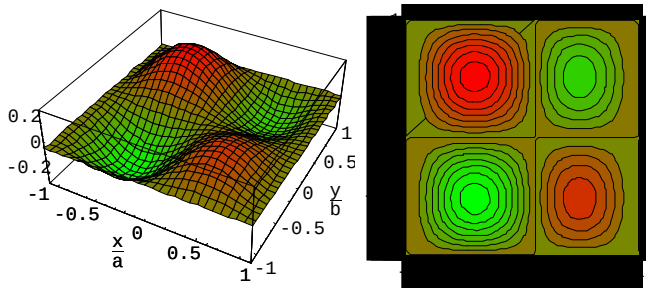


Figure 6b. Eigenshape for $P = 294.219$

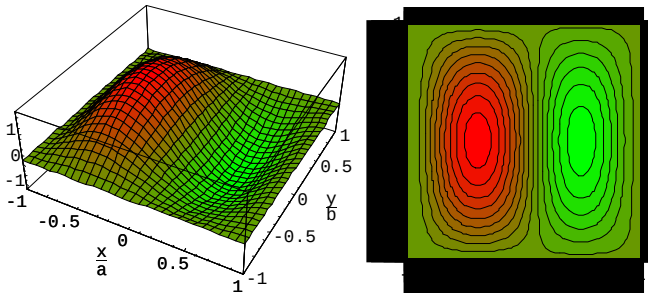


Figure 6c. Eigenshape for $P = 977.085$.

8. **Effect of higher order polynomials**

The value of the critical load parameter decreases and converges with an increase in the polynomial order of the trial functions:

$n = m$	Critical load parameter
4	120.234
5	45.3935
6	38.5191
7	38.5062

Table 1. The critical loads

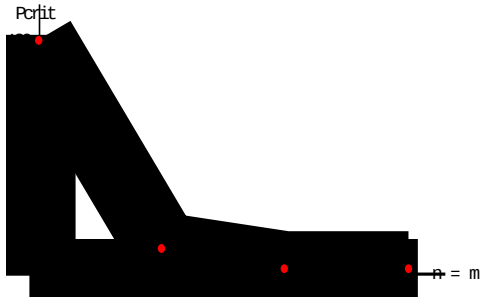


Figure 7. The critical load parameter as a function of the polynomial order of the trial functions

The first, critical eigenshapes are different but also converging to the true shape:

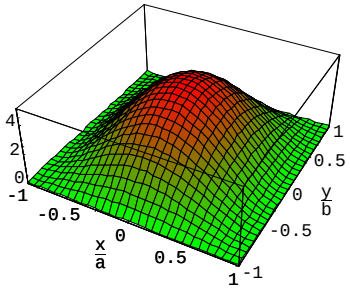


Figure 8a. Critical shape for $n = m = 4$

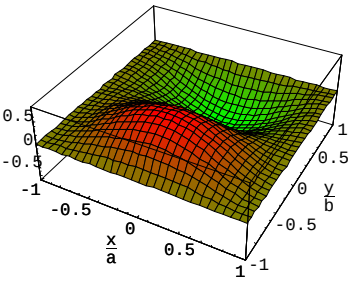


Figure 8b. Critical shape for $n = m = 5$

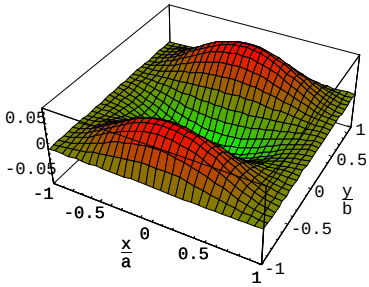


Figure 8c. Critical shape for $n = m = 6$

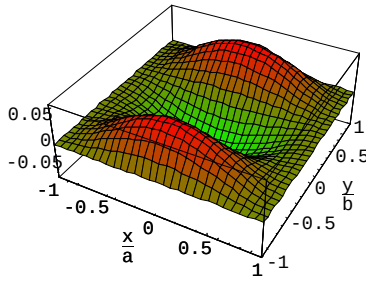


Figure 8d. Critical shape for $n = m = 7$

Employing higher order polynomials, new roots will emerge. For example, in case of $n = m = 6$, we obtain nine roots.

38.5191	45.3934	118.208	164.784	276.89
625.832	922.617	1203.99	4375.02	

Table 2. Roots for $n = m = 6$

The new roots result new eigenshapes. For example, for root 625.832, we obtain the following eigenshape

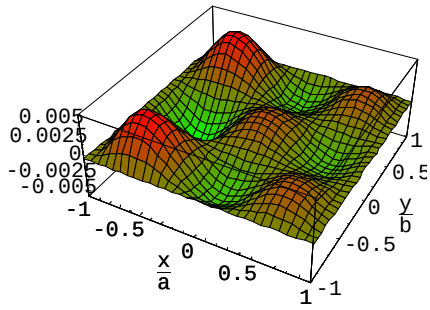


Figure 9. Eigenshape for root 625.832 in case of $n = m = 6$

A further increase of the order of the trial functions ($n, m \geqslant 7$) results in the appearance of complex and parasitic [7] (extremely big or small) roots.

9. **Concluding remarks**

A symbolic-numeric method has been developed to handle direct and inverse stability problems as a matrix eigenvalue problem. It was demonstrated that the direct stability problem can be considered to be a general, linear eigenvalue problem for the critical load parameter, and the inverse stability problem to be a nonlinear (quadratic) eigenvalue problem for a mechanical (stiffness) parameter. It has been shown that this method provides a simple, effective and elegant procedure for different boundary-value problems.

Acknowledgement. The supports provided by the Hungarian National Research Foundation T037880 and T025258 are gratefully acknowledged.

References

1. Huber, M.T.: *Die Theorie der kreuzweise bewehrten Eisenbetonplatten nebst Anwendungen auf mehrere bautechnisch wichtige Aufgaben über rechteckige Platten*, Bauingenieur 4, (1923), Heft 12, 354-360, Heft 13, 392-395.
2. Szabó, J.: *Ein Matrizenverfahren zur Berechnung von orthotropen stählernen Fahrbahnplatten*, Wissenschaflische Zeitschrift der Technischen Hochschule Dresden, **9**(3), (1960), 671-679.

3. Popper, Gy.: *Roots of polynomials with respesct to eigenvalue problems of theory of structures*, Dr. tech. Dissertation, Technical University of Budapest, 1967. (in Hungarian)
4. Beltzer, A.I.: *Engineering Analysis with Maple/Mathematica*, Academic Press, 1995.
5. Popper, Gy. and Gáspár, Zs.: *Numerical method for solving eigenvalue problems of a matrix of order n*, Mûszaki Tudomány, **57**, (1979), 49-56. (in Hungarian)
6. Wolfram, S.: *The Mathematica Book, Version 4*, Cambridge University Press, 1999.
7. Lánczos, C.: *Linear Differential Operators*, Dower Publications, 1997.

Journal of Computational and Applied Mechanics, Vol. 4., No. 1., (2003), pp. 53–70

ON INVERSE KINEMATICS AND KINETICS OF REDUNDANT SPACE MANIPULATOR SIMULATION

Bernd Schäfer, Rainer Krenn and Bernhard Rebele
Institute of Robotics and Mechatronics
German Aerospace Center (DLR), 82234 Wessling, Germany
bernd.schaefer@dlr.de
[Received: June 14, 2002]

Abstract. Redundant manipulators are nowadays favoured in several future space mission scenarios in order to enhance the skill and flexibility of the entire system. DLR, since many years, has been engaged in the development of light-weight robotic systems in modular design. Typical tasks for space applied robotics are to define robot kinematics and to calculate joint kinetics very rapidly in order to support the whole space mission design from the very beginning. Redundant manipulators then require the solution of the inverse kinematics problem for more than 6 degrees of freedom. Equivalently, the respective joint torques and forces are to be calculated by forward and backward recursions. Rather than applying conventional schemes based on pseudo-inverse matrix methods, we favour optimization with equality constraints, based upon the well-known Lagrange formalism. The optimization criteria are chosen to represent the underlying physical meaning, such as minimization of joint velocities, accelerations, torques or power, or even an optimization criterion that maintains the entire robot configuration during motion very close to a reference configuration. Simultaneously, this procedure also takes care that the joint loads and stresses in structural arm links do not exceed upper bounds. Two examples of light-weight robot design for space applications are presented that very clearly show the efficiency of the underlying algorithms.

Mathematical Subject Classification: 70B15
Keywords: inverse kinematics and kinetics, redundant manipulators, space robotics, optimization with constraints

1. Introduction

Future robotic-based satellite missions will be unimaginable without the demand for strong reduction of weight and for enhanced autonomy and intelligence. For many years DLR has paved the way for these challenging novel technologies. Currently, DLR is developing very light-weight robotic systems of its 3^{rd} generation in modular design, which are used as the design baseline for future space robot system scenarios of various kind [1]. Several studies are presently in progress (Figure 1) in order to prepare future robotized missions and to benefit from the use of robotic manipulators in space [2],[3]. It stands to reason that the desire to use light-weight and slender robotic systems is very strong since it promises very new application scenarios. Especially, this is more than ever true in all cases where the number of degrees of freedom

is increased beyond the traditional maximum number of six. Such redundant manipulators are able to enhance the skill and flexibility of the entire robotic system, and hence are able to increase the operational use and to obtain the maximum use of the robotic workspace. Moreover, avoiding collision with obstacles, and the avoidance of and passing through singularities are two more major aspects that make the use of redundancy in manipulators for space applications very preferable.

One of the prevailing design tasks for a given space mission scenario including robotics is the definition of the robot kinematics, as well as the number of joints and the sequence of joint axes orientations. Moreover, the knowledge of the joint kinetics states, i.e., forces and torques for given motions, is very important in order to rapidly support the whole space mission design from the very beginning. Modifications in manipulator design are then feasible at an early stage of the whole mission design in cases where upper bounds for manipulator loads and stresses are exceeded.

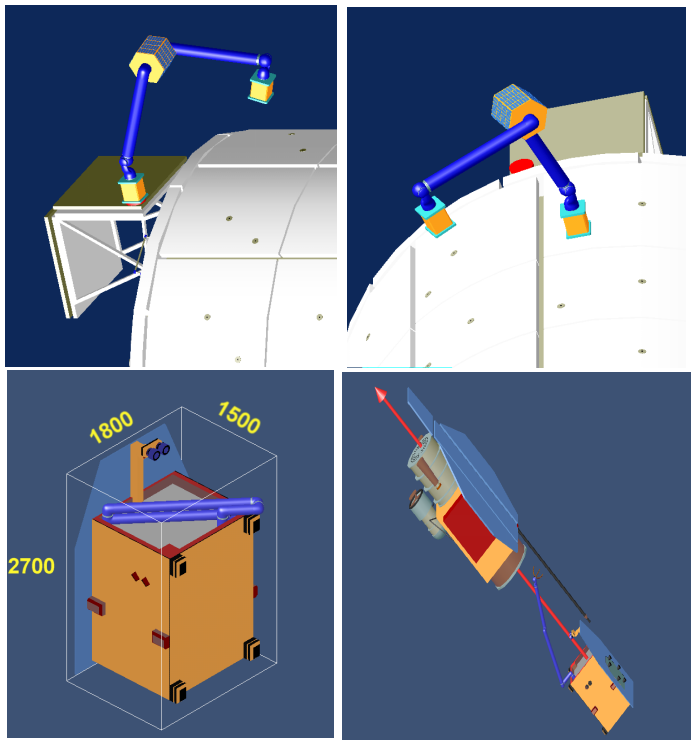


Figure 1. Two examples of near-future robotic-based space missions: MISSISS, Mobile Inspection and Service System for the International Space Station ISS (upper two figures), manipulator walk-over system operating on ISS Columbus external shell. Rescue satellite (lower two figures), manipulator used to capture a malfunctioning satellite (Rosat, the German Roentgen satellite, in orbit since 1990) followed by a de-orbiting scenario

However, the problem of solving the inverse kinematics increases according to the number of joints. Conventional solutions to this problem area are mostly based upon the use of pseudo-inverse matrix methods, where the physical meaning of the underlying method is not apparent at all. We have followed quite a different way for the inverse kinematics and kinetics problem solving. Currently, in our approach two different solutions are provided within our simulation environment, which both have their specific physical meanings. The first one is an optimization algorithm that minimizes, for example, the joint speed and acceleration subject to the end effector misalignment. The second one makes use of the differential equations describing the system of the robot motion whereby the robot is a passive chain of joints and limbs. For the latter one, the solution in joint coordinates is solved elegantly by the governing differential equations, rather than by solving algebraic equations and using minimization procedures, as in the first approach [2]. In this second case the motion is initiated by force and torque at the end effector and the joint motion can be calculated by solving the differential equations. Then, the dynamics parameters of the chain (e.g. joint friction, limb mass) affect the overall robot motion analogously to the optimization goal of the first method. This method is very appealing because it can be easily implemented in available multi-body dynamics simulation software. However, in this paper we will focus on the optimization method with equality constraints using Lagrange formalism for obtaining an optimal solution.

2. **Inverse kinematics and kinetics**

2.1. **Description of kinematics.** Figures 2 and 3 give a representation of the various coordinate systems and notations used to describe multi-body kinematics. The desired (index d within the formulas) end effector position in Cartesian coordinates is given by

$$\mathbf{r}_d = \begin{pmatrix} r_{d,x}, & r_{d,y}, & r_{d,z} \end{pmatrix}^{\mathrm{T}} \tag{1}$$

and the desired orientation of the end effector by

$$\mathbf{A}_d = \begin{pmatrix} A_{d,xx} & A_{d,xy} & A_{d,xz} \\ A_{d,yx} & A_{d,yy} & A_{d,yz} \\ A_{d,zx} & A_{d,zy} & A_{d,zz} \end{pmatrix}. \tag{2}$$

Then the inverse kinematics problem has to find a set of angular positions for rotary joints and a set of translational positions for prismatic joints (e.g. telescopic arms), which can be combined by a vector $\mathbf{q}$,

$$\mathbf{q} = \begin{pmatrix} q_1, & q_2, & \ldots, & q_n \end{pmatrix}^{\mathrm{T}} \tag{3}$$

with n degrees of freedom, and $n > 6$ for redundant manipulators. Both end effector position and orientation (hereafter called 'pose') can be combined by the homogeneous transformation matrix $\mathbf{B}_d$,

$$\mathbf{B}_d = \begin{pmatrix} A_{d,xx} & A_{d,xy} & A_{d,xz} & r_{d,x} \\ A_{d,yx} & A_{d,yy} & A_{d,yz} & r_{d,y} \\ A_{d,zx} & A_{d,zy} & A_{d,zz} & r_{d,z} \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{4}$$

leading finally to the problem

$$\mathbf{q} = f\left(\mathbf{B}_d\right) \tag{5}$$

to be solved. In the following, we will restrict our problem to rotary joints only. Unique solutions for the joint angles $\mathbf{q}$ exist only for $n = 6$, disregarding some special robot configurations like elbow up or down, etc.. For the general case of $n > 6$ we have to consider additional constraint equations that

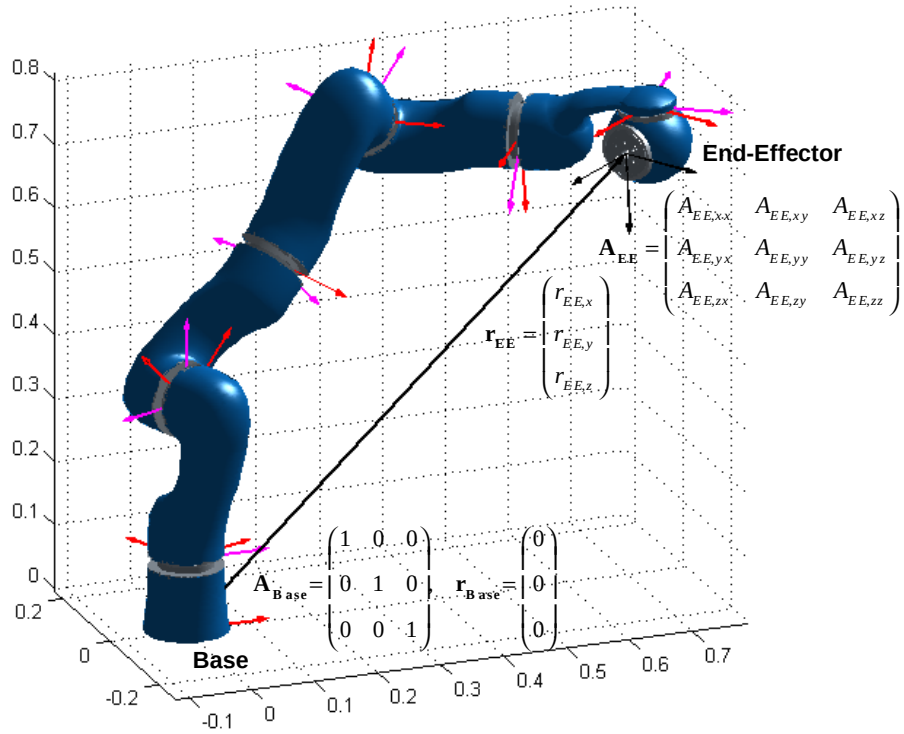


Figure 2. Kinematic structure and coordinate systems for the entire robot. The numbers at coordinates axes are in meters

guarantee a unique solution. We will not build upon common approaches used throughout the many examples for terrestrial applications, such as the well-known 'pseudo inverse method'. Rather, we make use of an optimization procedure with constraints based upon Lagrange formalism, which accounts for a real physical interpretation of the underlying method used here.

2.2. **Optimization based inverse kinematics.** The constraint given is that the solution of the inverse kinematics problem has to guarantee coincidence between the actual pose $\mathbf{B}$ and the desired pose $\mathbf{B}_d$. For mathematical reasons this deviation can be better represented by a homogenous transformation matrix $\mathbf{\Delta}$ (see e.g. [4])

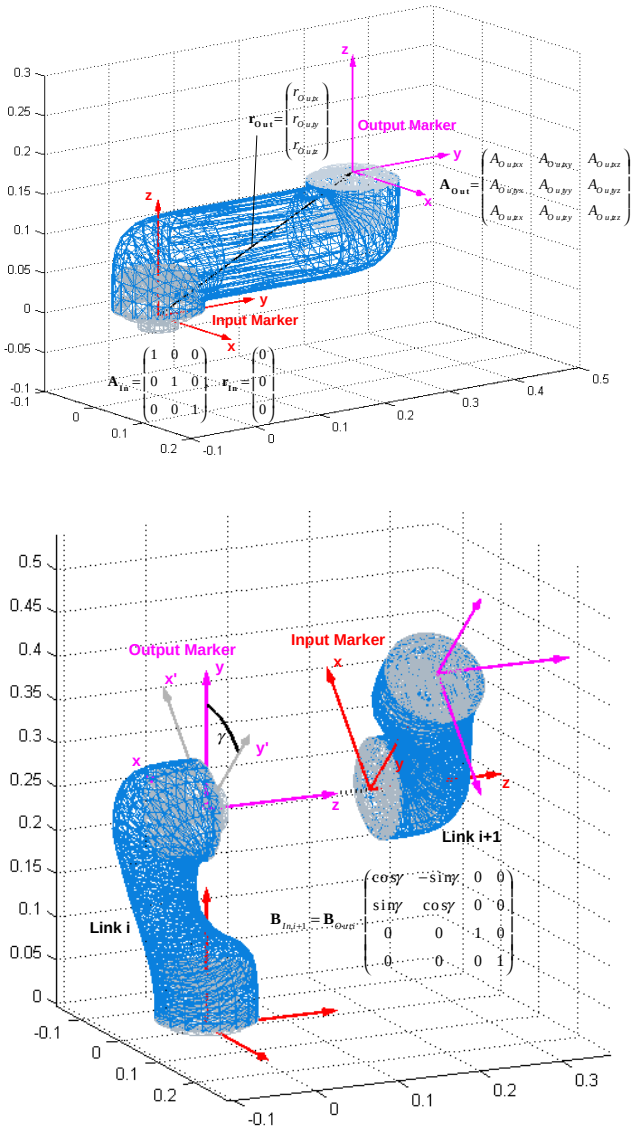


Figure 3. Coordinate systems for a single link and two adjacent joints with input and output marker. The numbers at coordinates axes are in meters

according to

$$\mathbf{\Delta} = \mathbf{B}_d^{-1} \cdot \mathbf{B} = \begin{pmatrix} \Delta_{11} & \Delta_{12} & \Delta_{13} & \Delta_{14} \\ \Delta_{21} & \Delta_{22} & \Delta_{23} & \Delta_{24} \\ \Delta_{31} & \Delta_{32} & \Delta_{33} & \Delta_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{6}$$

Should the desired pose be reached, then we will end up with $\mathbf{\Delta} = \mathbf{E}$, where $\mathbf{E}$ is the unity matrix. Then the problem to be solved can be transformed from matrix to vector form,

$$\mathbf{h} = \left(\Delta_{14}, \quad \Delta_{24}, \quad \Delta_{34}, \quad \Delta_{12} - \Delta_{21}, \quad \Delta_{23} - \Delta_{32}, \quad \Delta_{31} - \Delta_{13} \right)^{\mathrm{T}} = \mathbf{0}\,. \tag{7}$$

The first three elements are translational components whereas the last three represent rotational ones. This formulation of the inverse kinematics problem is free of singularities and therefore numerically stable. It can be shown [5] that for an incremental robot motion simulation the upper left 3x3 sub-matrix of $\mathbf{\Delta}$ will turn into a skew symmetric one. In the limit for small angles and positions going to zero, $\mathbf{\Delta}$ then goes to the identity matrix.

Considering the optimization criteria, a physical interpretation of the underlying method is favoured. Several different approaches can be followed, which may be distinguished by two kinds of goal functions f to be minimized: the first one is related to an optimization that focuses on joint drive kinematics and kinetics (joint drive oriented optimization), the second one takes care of the entire robot system configuration (configuration oriented optimization).

Joint drive oriented optimization criteria. The following criteria concern minimum loads and stresses applied to the robotic joints, and hence to the arm structures. For mathematical reasons, they are usually formulated as quadratic expressions of kinematic or kinetic states:

(a) Minimization of all the joint angular velocities $\dot{\mathbf{q}}$,

$$f_1 = \dot{\mathbf{q}}^{\mathrm{T}} \cdot \dot{\mathbf{q}}. \tag{8}$$

Mechanically, this may be interpreted as minimizing some linear damping force proportional to the joint velocities.

(b) Minimization of all the joint angular accelerations $\ddot{\mathbf{q}}$,

$$f_2 = \ddot{\mathbf{q}}^{\mathrm{T}} \cdot \ddot{\mathbf{q}}. \tag{9}$$

This optimization criterion may be favoured especially in cases where the robot is operates in a neighbourhood close to singularities, and where otherwise very high joint accelerations are required. This criterion will keep critical joints below their maximum load limits.

(c) Minimization of joint torques t,

$$f_3 = \mathbf{t}^{\mathrm{T}} \cdot \mathbf{t}. \tag{10}$$

(c) Minimization of power $P_i = t_i \omega_i$ consumed in each joint drive with number i,

$$f_4 = \mathbf{P}^{\mathrm{T}} \cdot \mathbf{P}\,, \tag{11}$$

where $\omega_i = \dot{q}_i$ and $\mathbf{P} = (P_1,\ P_2,\ ...,\ P_i,\ ...,\ P_n)^{\mathrm{T}}$ has been set. The latter two criteria require inverse kinetics calculation in order to obtain knowledge of the torques to be applied according to a given trajectory. Equations (8) to (11) also allow to weight each joint individually by a special weighting factor, in

order to be more flexible in assigning each single joint more or less importance during the optimization run.

Configuration oriented optimization criterion. The motivation for this criterion is to allow the robot a high degree of manoeuvrability. This means that during motion, each joint should keep its respective configuration far apart from its joint angle limits $q_{i,\max}$ and $q_{i,\min}$. This can also be defined by a reference angle according to $q_{ref,i} = (q_{i,\max} + q_{i,\min})/2$, whereas the robot will try to stay in a neighbourhood close to the desired configuration. Moreover, this reference position may also be defined by restrictions given by the environment where the robot is operating. A typical example is to prescribe a reference configuration for a collision-free motion within the robot's working space, which the robot has to follow very closely. As a minimization criterion, again a quadratic expression is given

$$f_5 \;=\; \Delta\mathbf{q}^{\mathrm{T}} \cdot \Delta\mathbf{q}\,, \tag{12}$$

where $\Delta\mathbf{q} = \mathbf{q} - \mathbf{q}_{ref}$ and $\mathbf{q}$ contains the actual angular positions of the joints. From a mechanical point of view, this procedure can be interpreted as minimizing stiffness forces acting in the joints, which originate in virtual springs somehow depending upon spring displacements $\Delta\mathbf{q}$.

Minimization of satellite base forces and torques. For space applications, the reaction forces and torques exerted on the satellite base by the manipulator motion, are in many cases undesirable or even jeopardize mission requirements. This is the case where micro-gravity conditions have to be maintained over a longer period. Minimizing the base torques and forces is therefore a primary goal for manipulator path and trajectory design.

In order to be be more general, all these criteria can be combined and appropriately weighted by special factors α_j

$$f \;=\; \sum \alpha_j\, f_j\,. \tag{13}$$

However, the best suited goal function has to be adapted to the respective application case, and cannot be generally determined.

Constraint optimization: Lagrange formalism. To solve the inverse kinematics problem, the well-known Lagrange formalism for constrained optimization is applied

$$L = f(\mathbf{q}) + \lambda^T \cdot \mathbf{h}(\mathbf{q}) = f(\mathbf{q}) + g(\mathbf{q}, \lambda)\,, \tag{14}$$

where L is the Lagrange function, f is the optimization criterion, $\mathbf{h}$ accounts for the equality constraints, and λ is a vector consisting of six Lagrange multipliers. This equation contains $n + 6$ unknowns, which can be combined by a vector $\mathbf{z}$,

$$\mathbf{z} = (q_1,\; q_2,\; \cdots,\; q_n,\; \lambda_1, \cdots, \lambda_6)^{\mathrm{T}} \tag{15}$$

giving finally $L = L(\mathbf{z})$. Minimization of L necessitates that the gradient ∇L goes to zero

$$\nabla L = \nabla \left(f(\mathbf{z}) + g(\mathbf{z}) \right) = \\ = \left(\frac{\partial f}{\partial q_1} + \frac{\partial g}{\partial q_1},\ \frac{\partial f}{\partial q_2} + \frac{\partial g}{\partial q_2},\ ...,\ \frac{\partial f}{\partial q_n} + \frac{\partial g}{\partial q_n},\ h_1,\ h_2,\ ...,\ h_6 \right)^{\mathrm{T}} \quad (16)$$

Equation (16) is a strongly non-linear system with $n + 6$ equations and unknowns. To solve it, the method based on an iteration according to Newton-Kantorowitsch (linearization by Taylor series expansion) may be favourably used. Given the Jacobi matrix

$$\mathbf{J} = \frac{\partial \mathbf{L}'}{\partial \mathbf{z}} \quad (17)$$

where $\mathbf{L}' = \nabla L$, a linearized system of $n + 6$ equations,

$$\mathbf{J}\left(\mathbf{z}^{(k)}\right) \cdot \Delta \mathbf{z}^{(k+1)} + \mathbf{L}'\left(\mathbf{z}^{(k)}\right) = \mathbf{0}, \qquad \text{and} \qquad \mathbf{z}^{(k+1)} = \mathbf{z}^{(k)} + \Delta \mathbf{z}^{(k+1)}, \quad (18)$$

is to be solved. The iteration will stop as soon as the deviation $\Delta \mathbf{z}^{(k+1)}$ remains below a given error limit. As initial value of $\mathbf{z}$, i.e. $\mathbf{z}^{(0)} = \mathbf{z}(t = 0)$, the current joint positions of the last, successfully terminated iteration step is taken, together with $\lambda^{(0)} = 0$.

2.3. **Inverse kinetics.** To solve the inverse kinetics problem, i.e., to obtain the forces and torques (kinetic states) acting in the joints, the dynamics description of the entire robot system, i.e., the differential equations of motion, has to be established. Based upon a multi-body system formulation, e.g. by Newton-Euler formulation, the joint motion $\mathrm{q}(t)$ and its time derivatives can be assigned to the applied forces and torques. Given the end effector trajectory, the joint kinematic states are found by the inverse kinematics problem, and the kinetic states are obtained by applying forward and backward calculations. The forward kinematics calculation gives the respective values required, i.e., the absolute velocities and accelerations of joints and the center of mass of each link), beginning from the inertially fixed frame up to the end effector position and orientation. Then, by backward calculation the actual forces and torques can be determined, beginning at the end effector level.

3. Basic example of a 7 DOF robot: Simulation results and discussion

The 7 dof (degrees of freedom) robot with fixed base of Figure 2 has been taken as a basic example to demonstrate the efficiency of the inverse kinematics and kinetics problem for redundant manipulators. The initial robot configuration is given in Figure 4 achieving a total length of about 1.25 m. The starting configuration is the one already shown in Figure 2, i.e., all joints have been moved by the same angle of 45^o. The end effector trajectory is prescribed: in our example we chose an elliptic planar path for the position within the three-dimensional working space, where the angle of the ellipse varies in time with a fifth order polynomial (see Figure 5 for the kinematic states). The orientation of the end effector is kept constant throughout

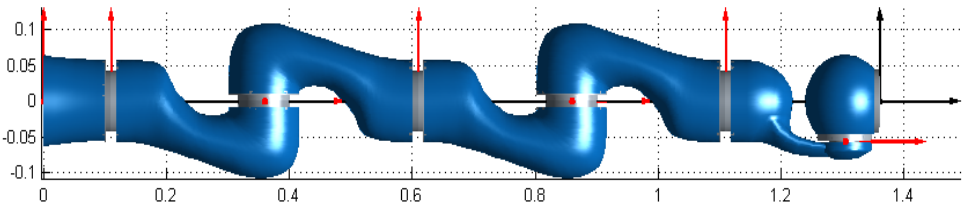


Figure 4. Initial robot configuration: 7 dof, kinematic configuration is (beginning from the fixed base) roll, pitch, roll, pitch, roll, pitch, roll; the numbers at both axes are in meters giving an overall length of about 1.25 m measured from the first roll axis

the elliptical motion. For comparison reasons, we have investigated three different optimization criteria: joint velocity, joint acceleration and configuration oriented minimization according to equations (8), (9) and (12). The two criteria based on kinetics states minimization, see equations (10) and (11), are still in the implementation phase. As the reference configuration for equation (12) we chose the starting configuration. It should be noted that in case of a non-redundant robot with 6 dof, the starting configuration of all the joints should be achieved again after terminating the elliptical motion (closed loop trajectory) of the end effector.

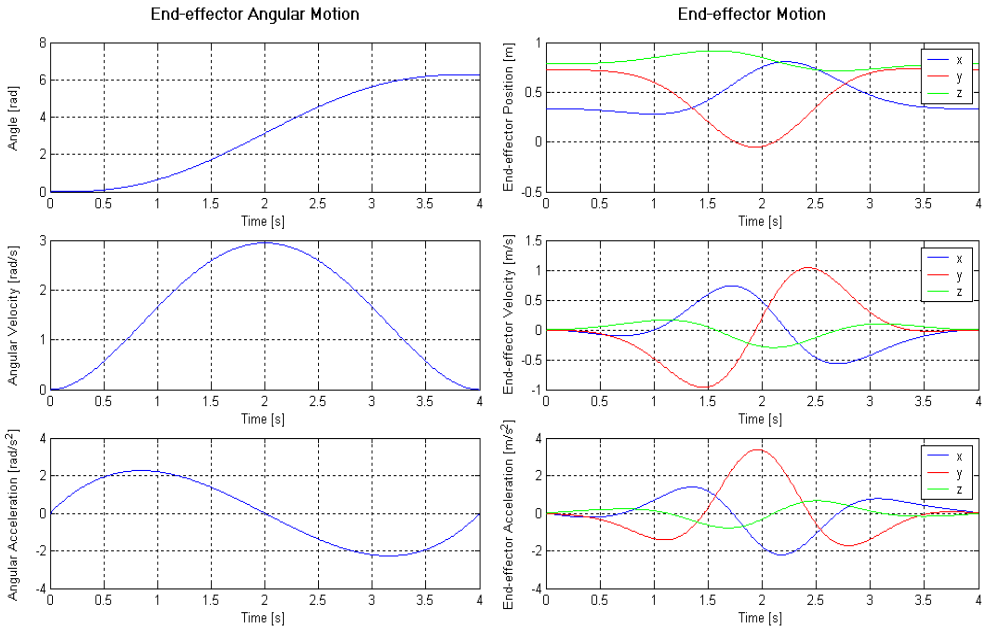


Figure 5. Kinematic states of the prescribed elliptical motion of the end effector

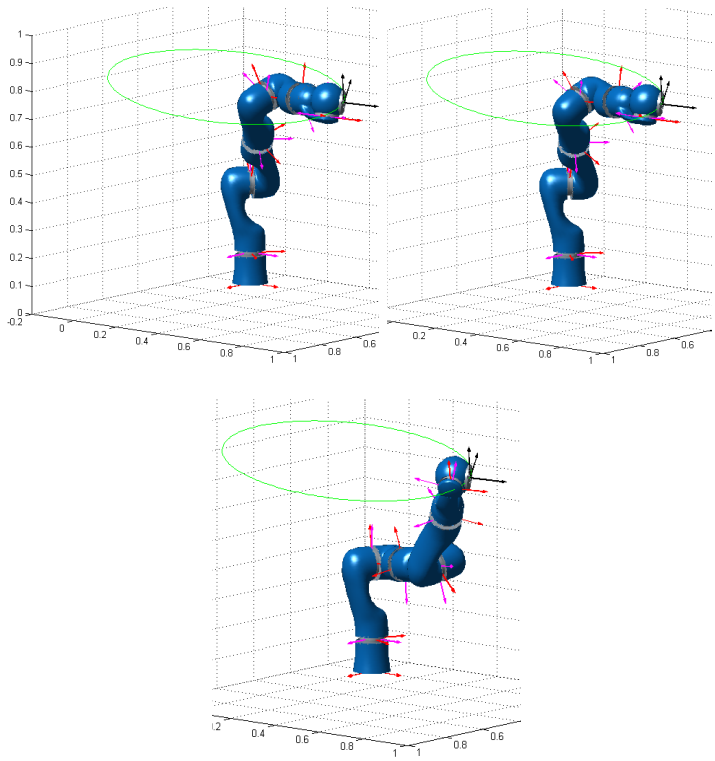


Figure 6. Final robot configuration achieved after one full elliptical loop: joint velocities optimized (left), reference configuration optimized (right), joint accelerations optimized (middle)

Figure 6 gives the comparison of the different final configurations achieved after one full elliptical loop. In the case of joint velocity and acceleration optimization we observe a great difference in the final configuration, whereas optimization of the reference, i.e. starting configuration does not exhibit a great deviation when compared to joint velocity minimization. Moreover, in the reference configuration case the final configuration is indeed identical to the initial, i.e., the reference configuration. The joint velocity optimization shows here some minor deviations from the initial configuration, whereas in joint acceleration minimization the significant joint angle deviations are surprising.

These differences in the kinematic states are presented graphically in Figure 7. Another striking behaviour in joint acceleration is to be noted: The joint velocities achieved after termination of the full elliptical trajectory are never zero. This is not surprising since the algorithm minimizes accelerations rather than velocities, although the joint accelerations achieved after trajectory termination are zero. For reasons of comparison, for each optimization criterion the respective corresponding kinematic state is presented graphically as well (Figure 8), together with the quadratic sum of the respective kinematic states.

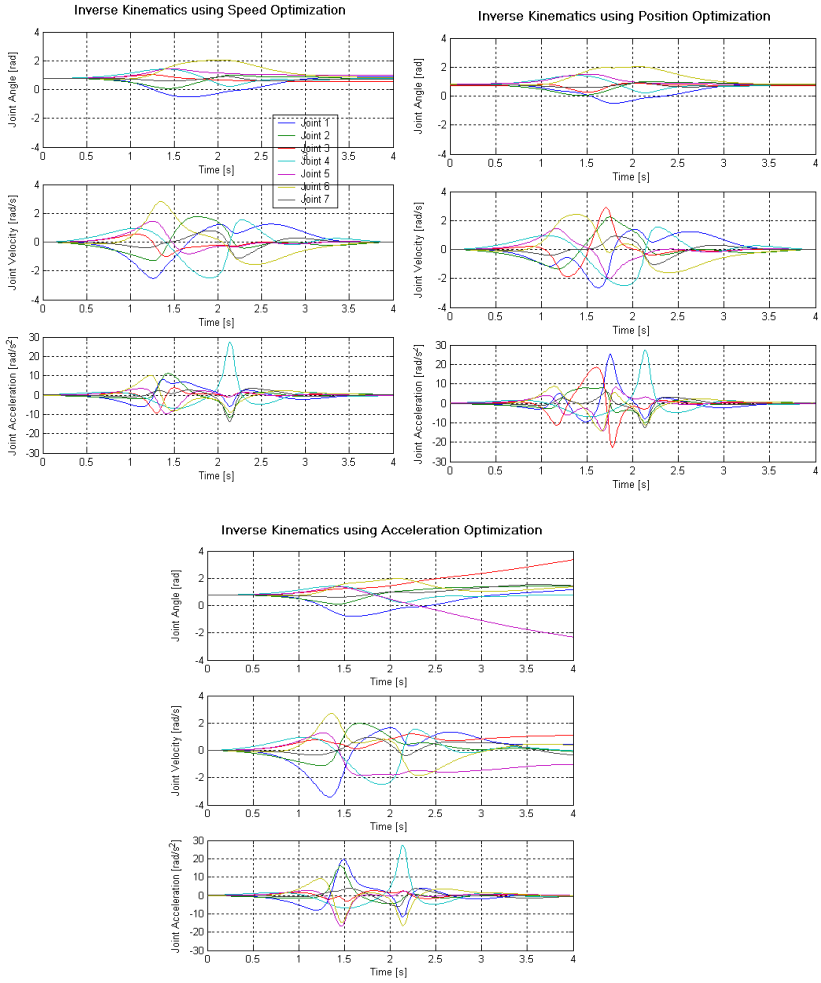


Figure 7. Kinematic states of each joint achieved by optimization

Finally, Figure 9 then shows an example of the calculated joint torque (w.r.t the joint axis) that is necessary in order to drive the respective joint. For the sake of compactness within this paper, the inertia parameters of each joint and link, which are required for the inverse kinetics calculation, are not presented within this paper.

Furthermore, instead of taking from a table the stored trajectory increments, a more common application in space robotics is to guide the robot by means of an input device, for example, a so-called 6-dof space ball. We call this mode of operations a tele-operated manipulation. This input device allows the human tele-operator to guide the robot end effector in 3 translational and in 3 rotational degrees of freedom. Again, the commanded trajectory given in some arbitrary manner by the human operator, is discretized by the space ball inherent software at a time

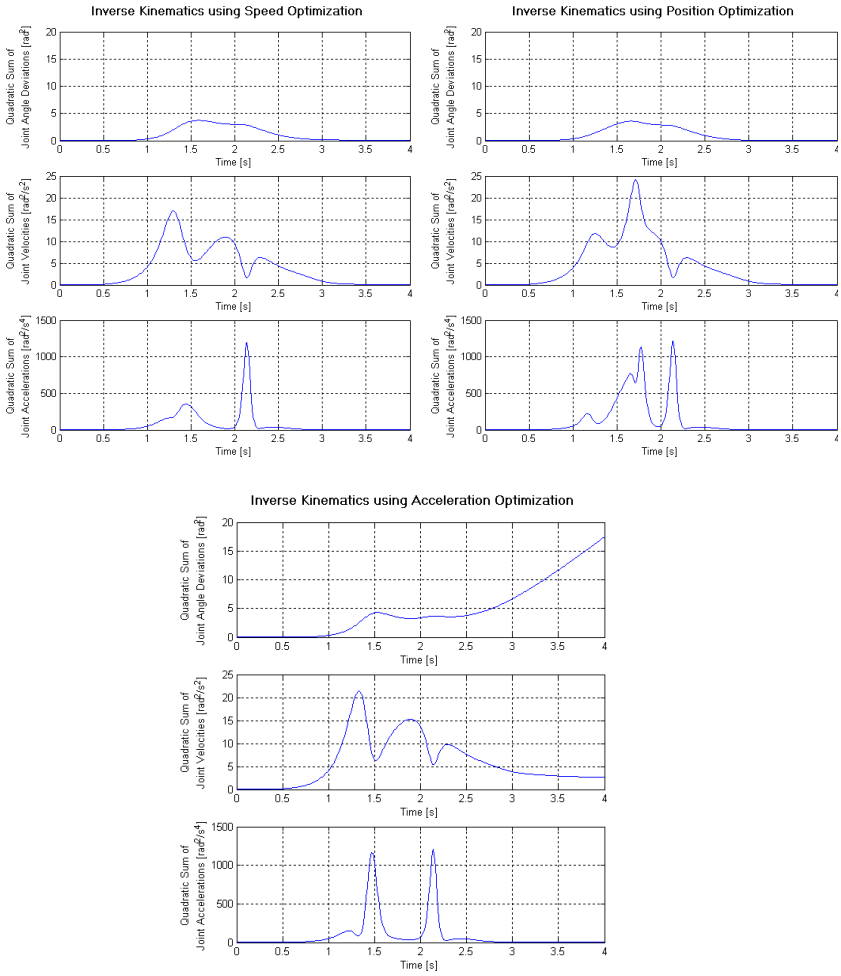


Figure 8. Quadratic sum of the respective kinematic joint states. Note, that in each optimization criterion only the respective quadratic sum at each time increment is optimal

interval equivalent or close to that of the table stored trajectory. Finally, for both kinds of robot guidance, i.e., either by a given trajectory stored in a table or by an input device based commanding via tele-operation, the underlying procedure for inverse kinematics and kinetics problem solving is highly efficient.

Final remarks have to be given to the optimization algorithm numerically implemented, and to the usefulness of the comparison of the different optimization criteria. The given trajectory is discretized by a fixed time interval of $T_s = 10$ ms. This interval is small enough to find a solution of the given inverse problem. Equivalently this means that the new prescribed end effector position and orientation does not change substantially from one time increment to the next. Otherwise, the procedure used

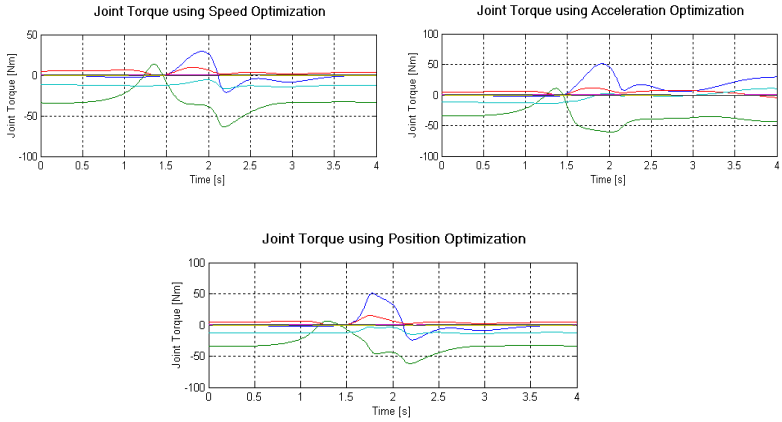


Figure 9. Calculated torques in each joint for the respective drive axis, for the different optimization criteria

here has to be modified appropriately (see specific example in Section 4), either by reducing the time interval or by changing the initial guesses for equation (18).

Concerning the comparison of results for different optimization criteria, it is found useless for two reasons. First, each optimization criterion naturally results in a configuration that is optimal in the given sense. The definition of any kind of measure with physical interpretation would be very helpful here. Second, the optimization based algorithm only gives an optimal result in terms of a new configuration that refers only to the configuration achieved in the time interval before. The reason is that the criterion is optimized only within the given time interval T_s. Therefore, the joint states are not optimized within the global time T, which covers the entire trajectory. This means, in general, that for the different optimization criteria used, the achieved new joint angles, velocities, accelerations (and torques, power, etc.) are different in nature for each time interval. Therefore, it makes no sense here to compare all criteria to each other in a sensible way.

However, a comparison may be considered sensible in the case where we let the optimization process cover the entire trajectory as a total. In that case, the optimization criteria defined in equations (8) to (12) have to be modified. For example, equations (8) then would read

$$f = \int_0^T \dot{\mathbf{q}}^T \dot{\mathbf{q}} \, \mathrm{d}t \tag{19}$$

and the optimization has to be accomplished off-line for the whole path.

Finally, all these individual optimization criteria can be combined while providing different weighting factors to each of them. Obviously, a best suited goal function cannot be given for optimization; rather it has to be adjusted to the specific application scenario. The results we obtained underline the efficiency of the algorithms being computationally very fast (1-2 ms for a complete inverse calculation within T_s, using

a personal computer with 1 GHz performance). More than ever, this last aspect is most important in all cases where need to perform an entire robotic motion simulation in real-time.

4. **Application: Robotics-based mission scenario**

To demonstrate the procedure for the inverse kinematics and kinetics calculation of redundant manipulators for a real scenario, a rescue satellite space mission is envisaged, according to Figure 1. Here, a large malfunctioning satellite (Rosat, the German Roentgen satellite in low earth orbit) is approached by a smaller rescue satellite that is equipped with a 2 m long manipulator of 7 dof. The very complex and delicate

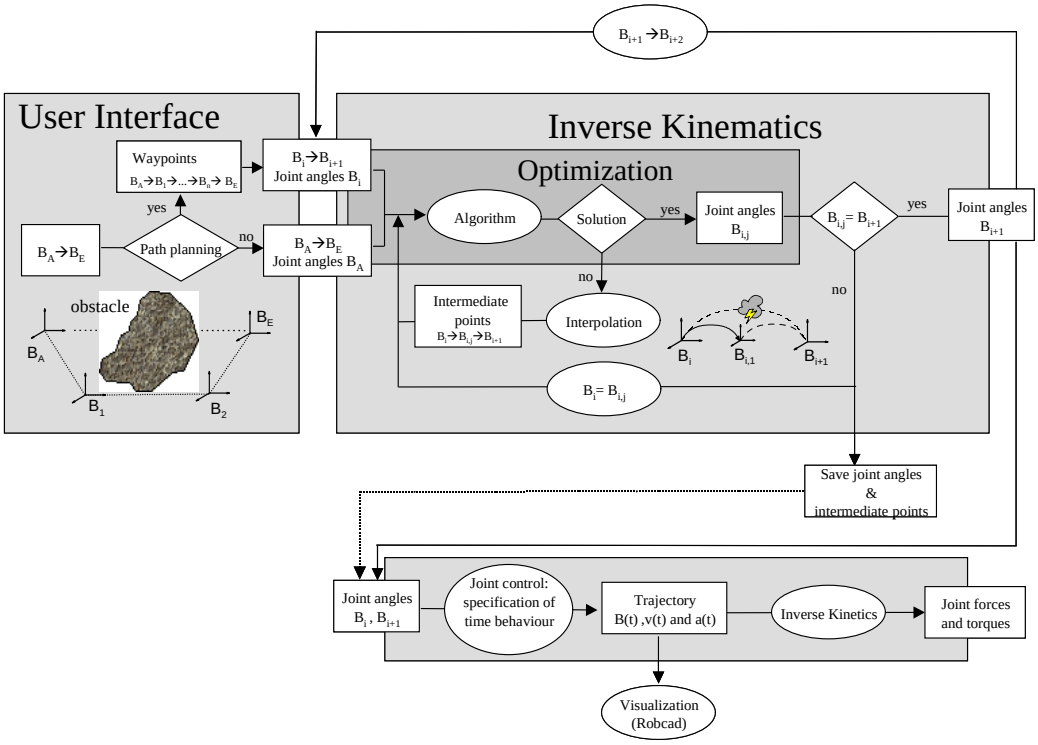


Figure 10. Structural diagram for computation of inverse kinematics and kinetics for fixed satellite base.

scenario comprises final approach, manipulator deployment, and finally grasping of the defective satellite by the manipulator. Our interest here is directed towards finding a suitable path for the manipulator's end effector, while avoiding any collision with obstacles in between. The approach and grasping strategy distinguishes between two kinds of operations: the first one assumes a fixed base of the manipulator, i.e., the rescue satellite is expected not to move during manipulator motion. This can

be guaranteed by means of the satellite's AOCS (attitude and orbit control system) being active. A second mode of operation supposes the rescue satellite's AOCS notto be active, leaving additional 6 degrees of freedom to the entire system, consisting of satellite and manipulator. Then our problem can be described as follows: the end effector starting point A is given assigned with an initial robot configuration with all joint angles given, and represented by the matrix $\mathbf{B}_A$ equivalent to equation (4). Also, the end point E to be reached is given, represented by the corresponding matrix $\mathbf{B}_E$, except for the respective joint angles that are not given. The overall task for the inverse kinematics problem then is to find the appropriate joint angles.

Applying first the procedure for the fixed satellite base, Figure 10 gives an idea how to solve this problem. Within the 'user interface' block the user defines whether to directly proceed from point A to E (direct path) or to define some so-called waypoints, which are appropriately chosen in between both points. The operator, by his experience or best knowledge about the entire system behaviour, may choose such points in cases where safety distances have to be maintained between manipulator and satellite parts, where obstacles have to be avoided or straight ahead of the final point to be grasped (in order to keep the manipulator in a halting mode before the final complex grasping operation is carried out). In kinematic description, these waypoints are equivalent to the final point E, i.e., the end effector position and orientation is represented by a given matrix $\mathbf{B}_i$, but the respective joint angles are not known.

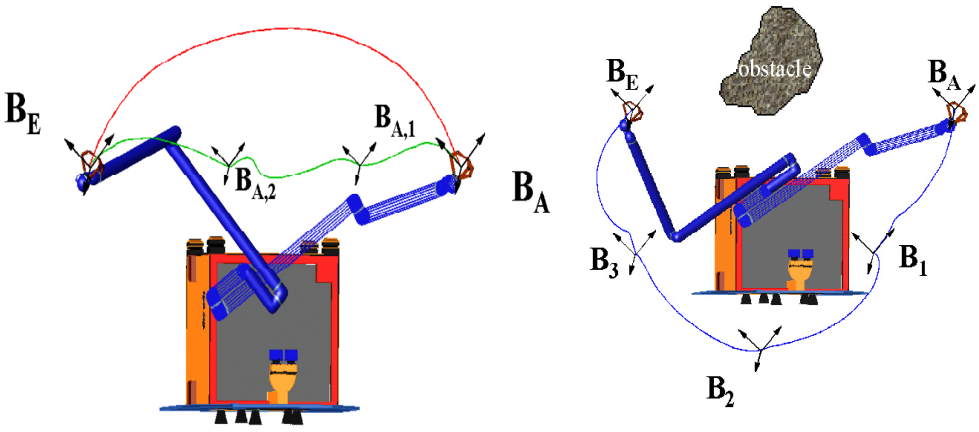


Figure 11. Left: Path and configuration for direct path (red line) or for allowing for intermediate points (green line). Right: Path and configuration for setting waypoints in between starting and end points. The 3 dimensional spatial paths are calculated using Robcad animation software package with built-in joint control algorithms. Note that although the starting configuration is the same in the left and right figures, this is not the case of the achieved final configurations, which are quite different

In general, the spatial distance between $\mathbf{B}_A$ and $\mathbf{B}_E$, or also between two neighbouring waypoints $\mathbf{B}_i$ and $\mathbf{B}_{i+1}$, is by all means long compared to the very closely spaced and discretized trajectory points of the basic example of Section 3. Furthermore, the underlying algorithm is based on linearization that is expected to give solutions more for local, closely distant points. Therefore, it is supposed that a solution is not achieved for all cases of two neighbouring waypoints a solution is achieved. This suggestion proved true during simulation runs, and a modified search strategy therefore has been set up. We decided, between two waypoints, to set so-called intermediate points $\mathbf{B}_{i,j}$ that guarantee an appropriate solution (Figure 10). The way to find such a point is to determine a search algorithm, arbitrarily chosen, that takes care of finding a suitable interpolation ratio fast and avoiding intermediate points near singularities. Therefore, between the given matrices B_i and B_{i+1} a linear interpolation is applied to find the cartesian coordinates and the Euler angles for the end effector.

Figure 11 presents the final results for the two choices, either to go directly from $\mathbf{B}_A$ to $\mathbf{B}_E$ (red path; or with intermediate points in between, green paths) or to go via waypoints $\mathbf{B}_1$, $\mathbf{B}_2$, $\mathbf{B}_3$ in order to avoid an obstacle for example (blue path). The path between two waypoints is not a straight line at all. In fact, it depends upon the joint control chosen, i.e., upon the specification of the time behaviour of the kinematic states $\mathbf{q}(t)$, $\dot{\mathbf{q}}(t)$ and $\ddot{\mathbf{q}}(t)$. Visualization and animation of the robot motions were performed by means of the software package Robcad (manufactured by Tecnomatix), which also provides the joint kinematics time specification (constant joint acceleration with upper bound as well as an upper bound for joint velocity).

Concerning the inverse kinematics and kinetics problem for a moving satellite base, the overall dynamics behaviour of the entire system has to be regarded. Figure 12 outlines the approach that will be followed for our further investigations. Since the manipulator motion causes a corresponding movement of the base, the algorithm for the fixed base has to be expanded by a part reflecting the influence of these additional 6 dof of the base, while taking the well-known laws for momentum and impulse also into consideration.

5. Concluding remarks and further activities

The inverse kinematics problem for redundant space manipulators has been solved by making use of the physical meaning of the underlying optimization criteria. Moreover, this method also accounts for the joint loads and stresses in structural arm links not to exceed upper bounds. Two different examples were presented, a more convential one for basic investigations of the quality of the approach, and a realistic one that simulates a typical space mission scenario where a robot attached to a rescue satellite is intended to grasp a second, malfunctioning satellite. The algorithms applied to both examples proved very efficient and they are computationally extremely fast. More than ever, the last aspect is most important in all cases where we have the need to perform an entire robotic motion simulation in real-time.

Should the inverse kinetics computation show that the joint torques and forces obtained are not below certain limits, then the robot design has to start again. This

means that especially the mechanical sub-components as motors, gears, and link structures, have to be changed according to a CAD based data panel, for example. This

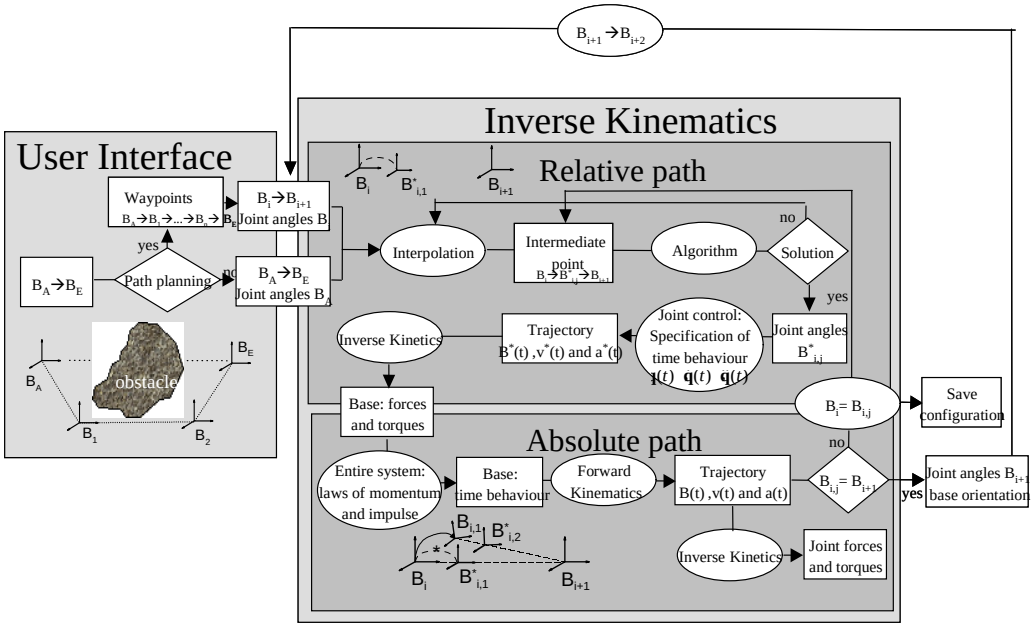


Figure 12. Structural diagram for computation of inverse kinematics and kinetics for moving satellite base.

starts the inverse kinematics and kinetics problem solving again. However, there are still more problems to be solved in further investigations. One refers to the moving satellite base, where we suggested an approach for the inverse problems. Another important item concerns an optimal trajectory planning, which we have not addressed here, but mentioned one possible approach as given by equation (19). Moreover, the addressed optimization criteria based on kinetics states minimization (e.g. joint torque and power) will be followed further.

References

1. Schäfer, B., Krenn, R., Hirzinger, G. and da Silva, A.R.: *Light-weight space robotics: rapid design approach and efficient simulation environment*, MIROC Machine Intelligence and Robotic Control, **3**(3), (2001), 99-111.
2. Krenn, R. and Schäfer, B.: *Dynamics simulation and assembly environment for rapid manipulator design*, GAMM Scientific Annual Meeting, Augsburg, Germany, 25-28 March 2002.
3. Rebele, B., Schäfer, B. and Borrell Lopez-Doriga, D.: *Dynamics aspects in satellite capturing by robotic manipulator*, GAMM Scientific Annual Meeting, Augsburg, Germany, 25-28 March 2002.

4. Weiss, H.: *Genetic Algorithms and Optimum Robot Design*, DLR Internal Report, October 2001.
5. Spong, M.W. and Vidyasagar, M.: *Robot Dynamics and Control*, John Wiley, New York, 1989.

Journal of Computational and Applied Mechanics, Vol. 4., No. 1., (2003), pp. 71–88

STABILIZATION PARAMETERS IN SUPG AND PSPG FORMULATIONS

Tayfun Tezduyar and Sunil Sathe
Mechanical Engineering, Rice University – MS 321
6100 Main Street, Houston, Texas 77005, USA
tezduyar@mems.rice.edu

[Received: June 29, 2002]

Abstract. We describe how we determine the stabilization parameters and element length scales used in the stabilized finite element formulations in fluid mechanics. These formulations include the interface-tracking and interface-capturing techniques we developed for computation of flow problems with moving boundaries and interfaces. The stabilized formulations we focus on are the streamline-upwind/Petrov-Galerkin (SUPG) and pressure-stabilizing/Petrov-Galerkin (PSPG) methods. The stabilization parameters described here are designed for the semi-discrete and space-time formulations of the advection-diffusion equation and the Navier-Stokes equations of incompressible flows.

Mathematical Subject Classification: 76M10
Keywords: stabilization parameters, element length scales, SUPG formulation, PSPG formulation, space-time formulation

1. Introduction

Most finite element techniques and computations reported in recent literature for computational fluid mechanics are based on stabilized formulations. The interface-tracking and interface-capturing techniques we have developed in recent years (see [1, 2, 3, 4]) for flows with moving boundaries and interfaces are also based on stabilized formulations. An interface-tracking technique, such as the Deforming-Spatial-Domain/Stabilized Space-Time (DSD/SST) formulation [1], requires meshes that "track" the interfaces. The mesh needs to be updated as the flow evolves. In interface-capturing techniques, such as one designed for two-fluid flows, the computations are based on spatial domains that are typically not moving or deforming. An interface function, marking the location of the interface, needs to be computed to "capture" the interface over the non-moving mesh.

The finite element flow techniques we have developed, including the interface-tracking and interface-capturing techniques, are based on the streamline-upwind/Petrov-Galerkin (SUPG) [5], Galerkin/least-squares (GLS) [6], and pressure-stabilizing/Petrov-Galerkin (PSPG) [1] formulations. In the interface-capturing techniques, stabilized semi-discrete formulations are used for both the Navier-Stokes equations of incompressible flows and the advection equation governing the time-evolution of an

interface function marking the interface location. These stabilization techniques prevent numerical oscillations and other instabilities in solving problems with advection-dominated flows and when using equal-order interpolation functions for velocity and pressure. In these stabilized formulations, judicious selection of the stabilization parameter, which is almost always known as "τ", plays an important role in determining the accuracy of the formulation. This stabilization parameter involves a measure of the local length scale (also known as "element length") and other parameters such as the local Reynolds and Courant numbers. Various element lengths and τs were proposed starting with those in [5] and [7], followed by the one introduced in [8], and those proposed in the subsequently reported SUPG, GLS and PSPG methods. A number of τs, dependent upon spatial and temporal discretizations, were introduced and tested in [9]. More recently, τs which are applicable to higher-order elements were proposed in [10].

Ways to calculate τs from the element-level matrices and vectors were first introduced in [11]. These new definitions are expressed in terms of the ratios of the norms of the relevant matrices or vectors. They take into account the local length scales, advection field and the element-level Reynolds number. Based on these definitions, a τ can be calculated for each element, or even for each element node or degree of freedom or element equation. Certain variations and complements of these new τs were introduced in [12, 4, 13]. In this paper, we describe the element-matrix-based and element-vector-based τs designed for the semi-discrete and space-time formulations of the advection-diffusion equation and the Navier-Stokes equations of incompressible flows. We also describe approximate versions of these τs that are based on the local length scales for the advection- and diffusion-dominated limits. In the test computations reported in Section 13, the performance of the stabilization parameters is evaluated for the advection-diffusion and Navier-Stokes equations.

2. Governing Equations

Let $\Omega_t \subset \mathbb{R}^{n_{sd}}$ be the spatial fluid mechanics domain with boundary Γ_t at time $t \in (0, T)$, where the subscript t indicates the time-dependence of the spatial domain. The Navier-Stokes equations of incompressible flows can be written on Ω_t and $\forall t \in (0, T)$ as

$$\rho(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla}\mathbf{u} - \mathbf{f}) - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} = 0, \tag{2.1}$$

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0, \tag{2.2}$$

where ρ, $\mathbf{u}$ and $\mathbf{f}$ are the density, velocity and the external force, and $\boldsymbol{\sigma}$ is the stress tensor:

$$\boldsymbol{\sigma}(p, \mathbf{u}) = -p\mathbf{I} + 2\mu\boldsymbol{\epsilon}(\mathbf{u}), \quad \boldsymbol{\epsilon}(\mathbf{u}) = \frac{1}{2}((\boldsymbol{\nabla}\mathbf{u}) + (\boldsymbol{\nabla}\mathbf{u})^T). \tag{2.3}$$

Here p is pressure, $\mathbf{I}$ is the identity tensor, $\mu = \rho\nu$ is viscosity, ν is the kinematic viscosity, and $\boldsymbol{\epsilon}(\mathbf{u})$ is the strain-rate tensor. The essential and natural boundary conditions for equation (2.1) are represented as

$$\mathbf{u} = \mathbf{g} \text{ on } (\Gamma_t)_g, \qquad \mathbf{n} \cdot \boldsymbol{\sigma} = \mathbf{h} \text{ on } (\Gamma_t)_h, \tag{2.4}$$

where $(\Gamma_t)_g$ and $(\Gamma_t)_h$ are complementary subsets of the boundary Γ_t, $\mathbf{n}$ is the unit normal vector, and $\mathbf{g}$ and $\mathbf{h}$ are given functions. A divergence-free velocity field $\mathbf{u}_0(\mathbf{x})$ is specified as the initial condition.

If the problem does not involve any moving boundaries or interfaces, the spatial domain does not need to change with respect to time, and the subscript t can be dropped from Ω_t and Γ_t. This might be the case even for flows with moving boundaries and interfaces, if in the formulation used the spatial domain is not defined to be the part of the space occupied by the fluid(s). For example, we can have a fixed spatial domain, and model the fluid-fluid interfaces by assuming that the domain is occupied by two immiscible fluids, A and B, with densities ρ_A and ρ_B and viscosities μ_A and μ_B. In modeling a free-surface problem where Fluid B is irrelevant, we assign a sufficiently low density to Fluid B. An interface function ϕ serves as a marker identifying Fluid A and B with the definition $\phi = \{1 \text{ for Fluid A and } 0 \text{ for Fluid B}\}$. The interface between the two fluids is approximated to be at $\phi = 0.5$. In this context, ρ and μ are defined as

$$\rho = \phi\rho_A + (1-\phi)\rho_B, \quad \mu = \phi\mu_A + (1-\phi)\mu_B. \tag{2.5}$$

The evolution of the interface function ϕ, and therefore the motion of the interface, is governed by a time-dependent advection equation, written on Ω and $\forall t \in (0,T)$ as

$$\frac{\partial\phi}{\partial t} + \mathbf{u}\cdot\boldsymbol{\nabla}\phi = 0. \tag{2.6}$$

As a generalization of equation (2.6), let us consider over a domain Ω with boundary Γ the following time-dependent advection-diffusion equation, written on Ω and $\forall t \in (0,T)$ as

$$\frac{\partial\phi}{\partial t} + \mathbf{u}\cdot\boldsymbol{\nabla}\phi - \boldsymbol{\nabla}\cdot(\nu\boldsymbol{\nabla}\phi) = 0, \tag{2.7}$$

where ϕ represents the quantity being transported (e.g., temperature, concentration), and ν is the diffusivity. The essential and natural boundary conditions associated with equation (2.7) are represented as

$$\phi = \mathrm{g} \text{ on } \Gamma_{\mathrm{g}}, \qquad \mathbf{n}\cdot\nu\boldsymbol{\nabla}\phi = \mathrm{h} \text{ on } \Gamma_{\mathrm{h}}. \tag{2.8}$$

A function $\phi_0(\mathbf{x})$ is specified as the initial condition.

3. Stabilized Formulation for Advection-Diffusion Equation

Let us assume that we have constructed some suitably-defined finite-dimensional trial solution and test function spaces $\mathcal{S}_\phi^h$ and $\mathcal{V}_\phi^h$. The stabilized finite element formulation of equation (2.7) can then be written as follows: find $\phi^h \in \mathcal{S}_\phi^h$ such that $\forall w^h \in \mathcal{V}_\phi^h$:

$$\begin{aligned}
&\int_\Omega w^h\left(\frac{\partial\phi^h}{\partial t} + \mathbf{u}^h\cdot\boldsymbol{\nabla}\phi^h\right)d\Omega + \int_\Omega \boldsymbol{\nabla}w^h\cdot\nu\boldsymbol{\nabla}\phi^h d\Omega - \int_{\Gamma_{\mathrm{h}}} w^h \mathrm{h}^h d\Gamma \\
+ &\sum_{e=1}^{n_{el}}\int_{\Omega^e}\tau_{\mathrm{SUPG}}\mathbf{u}^h\cdot\boldsymbol{\nabla}w^h\left(\frac{\partial\phi^h}{\partial t} + \mathbf{u}^h\cdot\boldsymbol{\nabla}\phi^h - \boldsymbol{\nabla}\cdot\left(\nu\boldsymbol{\nabla}\phi^h\right)\right)d\Omega = 0.
\end{aligned} \tag{3.1}$$

Here n_{el} is the number of elements, Ω^e is the element domain, and τ_{SUPG} is the SUPG stabilization parameter.

4. Element-Matrix-Based Stabilization Parameters for Advection-Diffusion Equation

Let us use the notation $\mathbf{b} : \int_{\Omega^e}(\ldots)d\Omega : \mathbf{b}_{\text{V}}$ to denote the element-level matrix $\mathbf{b}$ and element-level vector $\mathbf{b}_{\text{V}}$ corresponding to the element-level integration term $\int_{\Omega^e}(\ldots)d\Omega$. We define the following element-level matrices and vectors:

$$\mathbf{m}: \quad \int_{\Omega^e} w^h \frac{\partial \phi^h}{\partial t} d\Omega \quad : \mathbf{m}_{\text{V}}, \tag{4.1}$$

$$\mathbf{c}: \quad \int_{\Omega^e} w^h \mathbf{u}^h \cdot \boldsymbol{\nabla} \phi^h d\Omega \quad : \mathbf{c}_{\text{V}}, \tag{4.2}$$

$$\mathbf{k}: \quad \int_{\Omega^e} \boldsymbol{\nabla} w^h \cdot \nu \boldsymbol{\nabla} \phi^h d\Omega \quad : \mathbf{k}_{\text{V}}, \tag{4.3}$$

$$\tilde{\mathbf{k}}: \quad \int_{\Omega^e} \mathbf{u}^h \cdot \boldsymbol{\nabla} w^h \; \mathbf{u}^h \cdot \boldsymbol{\nabla} \phi^h d\Omega \quad : \tilde{\mathbf{k}}_{\text{V}}, \tag{4.4}$$

$$\tilde{\mathbf{c}}: \quad \int_{\Omega^e} \mathbf{u}^h \cdot \boldsymbol{\nabla} w^h \frac{\partial \phi^h}{\partial t} d\Omega \quad : \tilde{\mathbf{c}}_{\text{V}}. \tag{4.5}$$

We define the element-level Reynolds and Courant numbers as follows:

$$Re = \frac{\|\mathbf{u}^h\|^2}{\nu} \frac{\|\mathbf{c}\|}{\|\tilde{\mathbf{k}}\|}, \tag{4.6}$$

$$Cr_u = \frac{\Delta t}{2} \frac{\|\mathbf{c}\|}{\|\mathbf{m}\|}, \tag{4.7}$$

$$Cr_\nu = \frac{\Delta t}{2} \frac{\|\mathbf{k}\|}{\|\mathbf{m}\|}, \tag{4.8}$$

where $\|\mathbf{b}\|$ is the norm of matrix $\mathbf{b}$.

The components of element-matrix-based τ_{SUPG} are defined as follows:

$$\tau_{\text{S1}} = \frac{\|\mathbf{c}\|}{\|\tilde{\mathbf{k}}\|},$$

$$\tau_{\text{S2}} = \frac{\Delta t}{2} \frac{\|\mathbf{c}\|}{\|\tilde{\mathbf{c}}\|}, \tag{4.9}$$

$$\tau_{\text{S3}} = \tau_{\text{S1}} Re = \left(\frac{\|\mathbf{c}\|}{\|\tilde{\mathbf{k}}\|} \right) Re. \tag{4.10}$$

To construct τ_{SUPG} from its components we proposed in [11] the form

$$\tau_{\text{SUPG}} = \left(\frac{1}{\tau_{\text{S1}}^r} + \frac{1}{\tau_{\text{S2}}^r} + \frac{1}{\tau_{\text{S3}}^r} \right)^{-\frac{1}{r}}. \tag{4.11}$$

The components of the element-vector-based τ_{SUPG} are defined as follows:

$$\tau_{\text{SV1}} = \frac{\|\mathbf{c}_{\text{V}}\|}{\|\tilde{\mathbf{k}}_{\text{V}}\|}, \tag{4.12}$$

$$\tau_{\text{SV2}} = \frac{\|\mathbf{c}_{\text{V}}\|}{\|\tilde{\mathbf{c}}_{\text{V}}\|}, \tag{4.13}$$

$$\tau_{\text{SV3}} = \tau_{\text{SV1}} Re = \left(\frac{\|\mathbf{c}_{\text{V}}\|}{\|\tilde{\mathbf{k}}_{\text{V}}\|}\right) Re. \tag{4.14}$$

With these three components,

$$(\tau_{\text{SUPG}})_{\text{V}} = \left(\frac{1}{\tau_{\text{SV1}}^r} + \frac{1}{\tau_{\text{SV2}}^r} + \frac{1}{\tau_{\text{SV3}}^r}\right)^{-\frac{1}{r}}. \tag{4.15}$$

5. Stabilized Formulation for Navier-Stokes Equations

Let us assume that we have some suitably-defined finite-dimensional trial solution and test function spaces for velocity and pressure: $\mathcal{S}_{\mathbf{u}}^h$, $\mathcal{V}_{\mathbf{u}}^h$, $\mathcal{S}_p^h$ and $\mathcal{V}_p^h = \mathcal{S}_p^h$. The stabilized finite element formulation of equations (2.1)-(2.2) can then be written as follows: find $\mathbf{u}^h \in \mathcal{S}_{\mathbf{u}}^h$ and $p^h \in \mathcal{S}_p^h$ such that $\forall \mathbf{w}^h \in \mathcal{V}_{\mathbf{u}}^h$ and $q^h \in \mathcal{V}_p^h$:

$$\begin{aligned}
&\int_\Omega \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla}\mathbf{u}^h - \mathbf{f}\right) d\Omega + \int_\Omega \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(p^h, \mathbf{u}^h) d\Omega - \int_{\Gamma_{\text{h}}} \mathbf{w}^h \cdot \mathbf{h}^h d\Gamma \\
+ &\int_\Omega q^h \boldsymbol{\nabla} \cdot \mathbf{u}^h d\Omega + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \frac{1}{\rho} [\tau_{\text{SUPG}} \rho \mathbf{u}^h \cdot \nabla \mathbf{w}^h + \tau_{\text{PSPG}} \nabla q^h] \cdot \\
&\qquad \left[\rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla}\mathbf{u}^h\right) - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}(p^h, \mathbf{u}^h) - \rho \mathbf{f}\right] d\Omega \\
+ &\sum_{e=1}^{n_{el}} \int_{\Omega^e} \tau_{\text{LSIC}} \boldsymbol{\nabla} \cdot \mathbf{w}^h \rho \boldsymbol{\nabla} \cdot \mathbf{u}^h d\Omega \;=\; 0.
\end{aligned} \tag{5.1}$$

Here τ_{PSPG} and τ_{LSIC} are the PSPG and LSIC (least-squares on incompressibility constraint) stabilization parameters.

6. Element-Matrix-Based Stabilization Parameters for Navier-Stokes Equation

We define the following element-level matrices and vectors:

$$\mathbf{m}: \quad \int_{\Omega^e} \mathbf{w}^h \cdot \rho \frac{\partial \mathbf{u}^h}{\partial t} d\Omega \quad : \mathbf{m}_{\text{V}}, \tag{6.1}$$

$$\mathbf{c}: \quad \int_{\Omega^e} \mathbf{w}^h \cdot \rho (\mathbf{u}^h \cdot \boldsymbol{\nabla}\mathbf{u}^h) d\Omega \quad : \mathbf{c}_{\text{V}}, \tag{6.2}$$

$$\mathbf{k}: \quad \int_{\Omega^e} \boldsymbol{\varepsilon}(\mathbf{w}^h) : 2\mu\boldsymbol{\varepsilon}(\mathbf{u}^h) d\Omega \quad : \mathbf{k}_{\mathrm{V}}, \tag{6.3}$$

$$\mathbf{g}: \quad \int_{\Omega^e} (\boldsymbol{\nabla} \cdot \mathbf{w}^h) p^h d\Omega \quad : \mathbf{g}_{\mathrm{V}}, \tag{6.4}$$

$$\mathbf{g}^{\mathrm{T}}: \quad \int_{\Omega^e} q^h (\boldsymbol{\nabla} \cdot \mathbf{u}^h) d\Omega \quad : \mathbf{g}^{\mathrm{T}}_{\mathrm{V}}, \tag{6.5}$$

$$\tilde{\mathbf{k}}: \quad \int_{\Omega^e} (\mathbf{u}^h \cdot \boldsymbol{\nabla}\mathbf{w}^h) \cdot \rho(\mathbf{u}^h \cdot \boldsymbol{\nabla}\mathbf{u}^h) d\Omega \quad : \tilde{\mathbf{k}}_{\mathrm{V}}, \tag{6.6}$$

$$\tilde{\mathbf{c}}: \quad \int_{\Omega^e} (\mathbf{u}^h \cdot \boldsymbol{\nabla}\mathbf{w}^h) \cdot \rho \frac{\partial \mathbf{u}^h}{\partial t} d\Omega \quad : \tilde{\mathbf{c}}_{\mathrm{V}}, \tag{6.7}$$

$$\tilde{\gamma}: \quad \int_{\Omega^e} (\mathbf{u}^h \cdot \boldsymbol{\nabla}\mathbf{w}^h) \cdot \boldsymbol{\nabla} p^h d\Omega \quad : \tilde{\gamma}_{\mathrm{V}}, \tag{6.8}$$

$$\beta: \quad \int_{\Omega^e} \boldsymbol{\nabla} q^h \cdot \frac{\partial \mathbf{u}^h}{\partial t} d\Omega \quad : \beta_{\mathrm{V}}, \tag{6.9}$$

$$\gamma: \quad \int_{\Omega^e} \boldsymbol{\nabla} q^h \cdot (\mathbf{u}^h \cdot \boldsymbol{\nabla}\mathbf{u}^h) d\Omega \quad : \gamma_{\mathrm{V}}, \tag{6.10}$$

$$\theta: \quad \int_{\Omega^e} \boldsymbol{\nabla} q^h \cdot \boldsymbol{\nabla} p^h d\Omega \quad : \theta_{\mathrm{V}}, \tag{6.11}$$

$$\mathbf{e}: \quad \int_{\Omega^e} (\boldsymbol{\nabla} \cdot \mathbf{w}^h) \rho (\boldsymbol{\nabla} \cdot \mathbf{u}^h) d\Omega \quad : \mathbf{e}_{\mathrm{V}}. \tag{6.12}$$

The element-level Reynolds and Courant numbers are defined the same way as they were defined before, as given by equations (4.6)-(4.8). The components of the element-matrix-based τ_{SUPG} are defined the same way as they were defined before, as given by equations (4)-(4.10). τ_{SUPG} is constructed from its components the same way as it was constructed before, as given by equation (4.11). The components of the element-vector-based τ_{SUPG} are defined the same way as they were defined before, as given by equations (4.12)-(4.14). The construction of $(\tau_{\mathrm{SUPG}})_{\mathrm{V}}$ is also the same as it was before, given by equation (4.15).

The components of the element-matrix-based τ_{PSPG} are defined as follows:

$$\tau_{\mathrm{P1}} = \frac{\|\mathbf{g}^{\mathrm{T}}\|}{\|\gamma\|}, \tag{6.13}$$

$$\tau_{\mathrm{P2}} = \frac{\Delta t}{2} \frac{\|\mathbf{g}^{\mathrm{T}}\|}{\|\beta\|}, \tag{6.14}$$

$$\tau_{\mathrm{P3}} = \tau_{\mathrm{P1}} Re = \left(\frac{\|\mathbf{g}^{\mathrm{T}}\|}{\|\gamma\|} \right) Re. \tag{6.15}$$

τ_{PSPG} is constructed from its components as follows:

$$\tau_{\mathrm{PSPG}} = \left(\frac{1}{\tau^r_{\mathrm{P1}}} + \frac{1}{\tau^r_{\mathrm{P2}}} + \frac{1}{\tau^r_{\mathrm{P3}}} \right)^{-\frac{1}{r}}. \tag{6.16}$$

The components of the element-vector-based τ_{PSPG} are defined as follows:

$$\tau_{\mathrm{PV1}} = \tau_{\mathrm{P1}}, \tag{6.17}$$

$$\tau_{\mathrm{PV2}} = \tau_{\mathrm{PV1}} \frac{\|\gamma_{\mathrm{V}}\|}{\|\beta_{\mathrm{V}}\|}, \tag{6.18}$$

$$\tau_{\mathrm{PV3}} = \tau_{\mathrm{PV1}} Re. \tag{6.19}$$

With these components,

$$(\tau_{\mathrm{PSPG}})_{\mathrm{V}} = \left(\frac{1}{\tau_{\mathrm{PV1}}^r} + \frac{1}{\tau_{\mathrm{PV2}}^r} + \frac{1}{\tau_{\mathrm{PV3}}^r} \right)^{-\frac{1}{r}}. \tag{6.20}$$

The element-matrix-based τ_{LSIC} is defined as follows:

$$\tau_{\mathrm{LSIC}} = \frac{\|\mathbf{c}\|}{\|\mathbf{e}\|}. \tag{6.21}$$

We define the element-vector-based τ_{LSIC} as:

$$(\tau_{\mathrm{LSIC}})_{\mathrm{V}} = \tau_{\mathrm{LSIC}}. \tag{6.22}$$

7. UGN-Based Stabilization Parameters for Navier-Stokes Equations

For the purpose of comparison, we define here also the stabilization parameters that are based on an earlier definition of the length scale h [8]:

$$h_{\mathrm{UGN}} = 2\, \|\mathbf{u}^h\| \left(\sum_{a=1}^{n_{en}} |\mathbf{u}^h \cdot \boldsymbol{\nabla} N_a| \right)^{-1}, \tag{7.1}$$

where N_a is the interpolation function associated with node a. The stabilization parameters are defined as follows:

$$\tau_{\mathrm{SUGN1}} = \frac{h_{\mathrm{UGN}}}{2\|\mathbf{u}^h\|}, \tag{7.2}$$

$$\tau_{\mathrm{SUGN2}} = \frac{\Delta t}{2}, \tag{7.3}$$

$$\tau_{\mathrm{SUGN3}} = \frac{h_{\mathrm{UGN}}^2}{4\nu}, \tag{7.4}$$

$$(\tau_{\mathrm{SUPG}})_{\mathrm{UGN}} = \left(\frac{1}{\tau_{\mathrm{SUGN1}}^2} + \frac{1}{\tau_{\mathrm{SUGN2}}^2} + \frac{1}{\tau_{\mathrm{SUGN3}}^2} \right)^{-\frac{1}{2}}, \tag{7.5}$$

$$(\tau_{\mathrm{PSPG}})_{\mathrm{UGN}} = (\tau_{\mathrm{SUPG}})_{\mathrm{UGN}}, \tag{7.6}$$

$$(\tau_{\mathrm{LSIC}})_{\mathrm{UGN}} = \frac{h_{\mathrm{UGN}}}{2} \|\mathbf{u}^h\|\, z. \tag{7.7}$$

Here z is given as follows:

$$z = \begin{cases} \left(\frac{Re_{\mathrm{UGN}}}{3}\right) & Re_{\mathrm{UGN}} \leq 3, \\ 1 & Re_{\mathrm{UGN}} > 3, \end{cases} \tag{7.8}$$

where $Re_{\mathrm{UGN}} = \frac{\|\mathbf{u}^h\| h_{\mathrm{UGN}}}{2\nu}$.

Comparisons between the performances of these earlier stabilization parameters and the ones proposed here can be found in [11]. These comparisons show that, especially for special element geometries, the performances are similar.

It was pointed out in [13] that the expression for τ_{SUGN1} can be written more directly as

$$\tau_{\text{SUGN1}} = \left(\sum_{a=1}^{n_{en}} |\mathbf{u}^h \cdot \boldsymbol{\nabla} N_a| \right)^{-1}, \tag{7.9}$$

and based on that, the expression for h_{UGN} can be written as

$$h_{\text{UGN}} = 2\|\mathbf{u}^h\| \, \tau_{\text{SUGN1}}. \tag{7.10}$$

A rationale for τ_{SUGN1} given by equation (7.9) was also provided in [13].

8. **Discontinuity-Capturing Directional Dissipation (DCDD)**

As a potential alternative or complement to the LSIC stabilization, we proposed in [12] the Discontinuity-Capturing Directional Dissipation (DCDD) stabilization. In describing the DCDD stabilization, we first define the unit vectors $\mathbf{s}$ and $\mathbf{r}$:

$$\mathbf{s} = \frac{\mathbf{u}^h}{\|\mathbf{u}^h\|}, \qquad \mathbf{r} = \frac{\boldsymbol{\nabla}\|\mathbf{u}^h\|}{\| \boldsymbol{\nabla}\|\mathbf{u}^h\| \, \|}, \tag{8.1}$$

and the element-level matrices and vectors $\mathbf{c}_{\text{r}}$, $\tilde{\mathbf{k}}_{\text{r}}$, $(\mathbf{c}_{\text{r}})_{\text{V}}$, and $(\tilde{\mathbf{k}}_{\text{r}})_{\text{V}}$:

$$\mathbf{c}_{\text{r}} : \qquad \int_{\Omega^e} \mathbf{w}^h \cdot \rho(\mathbf{r} \cdot \boldsymbol{\nabla}\mathbf{u}^h) d\Omega \qquad : (\mathbf{c}_{\text{r}})_{\text{V}} \ , \tag{8.2}$$

$$\tilde{\mathbf{k}}_{\text{r}} : \quad \int_{\Omega^e} (\mathbf{r} \cdot \boldsymbol{\nabla}\mathbf{w}^h) \cdot \rho(\mathbf{r} \cdot \boldsymbol{\nabla}\mathbf{u}^h) d\Omega \quad : (\tilde{\mathbf{k}}_{\text{r}})_{\text{V}} \ . \tag{8.3}$$

Then the DCDD stabilization is defined as

$$S_{\text{DCDD}} = \sum_{e=1}^{n_{el}} \int_{\Omega^e} \rho \nu_{\text{DCDD}} \boldsymbol{\nabla}\mathbf{w}^h : \left(\left[\mathbf{rr} - (\mathbf{r} \cdot \mathbf{s})^2 \mathbf{ss} \right] \cdot \boldsymbol{\nabla}\mathbf{u}^h \right) d\Omega, \tag{8.4}$$

where the element-matrix-based and element-vector-based DCDD viscosities are:

$$\nu_{\text{DCDD}} \quad = \quad |\mathbf{r} \cdot \mathbf{u}^h| \frac{\|\mathbf{c}_{\text{r}}\|}{\|\tilde{\mathbf{k}}_{\text{r}}\|} \ , \tag{8.5}$$

$$(\nu_{\text{DCDD}})_{\text{V}} \quad = \quad |\mathbf{r} \cdot \mathbf{u}^h| \frac{\|(\mathbf{c}_{\text{r}})_{\text{V}}\|}{\|(\tilde{\mathbf{k}}_{\text{r}})_{\text{V}}\|} \ . \tag{8.6}$$

An approximate version of the expression given by equation (8.5) can be written as

$$\nu_{\text{DCDD}} = |\mathbf{r} \cdot \mathbf{u}^h| \frac{h_{\text{RGN}}}{2}, \tag{8.7}$$

where

$$h_{\text{RGN}} = 2 \left(\sum_{a=1}^{n_{en}} |\mathbf{r} \cdot \boldsymbol{\nabla} N_a| \right)^{-1} . \tag{8.8}$$

A different way of determining ν_{DCDD} can be expressed as

$$\nu_{\text{DCDD}} = \tau_{\text{DCDD}} \|\mathbf{u}^h\|^2, \tag{8.9}$$

where

$$\tau_{\text{DCDD}} = \frac{h_{\text{DCDD}}}{2\|\mathbf{U}\|} \frac{\| \boldsymbol{\nabla}\|\mathbf{u}^h\| \| h_{\text{DCDD}}}{\|\mathbf{U}\|} . \tag{8.10}$$

Here $\mathbf{U}$ represents a global velocity scale, and h_{DCDD} can be calculated by using the expression

$$h_{\text{DCDD}} = 2 \frac{\|\mathbf{c}_{\text{r}}\|}{\|\tilde{\mathbf{k}}_{\text{r}}\|} , \tag{8.11}$$

or the approximation

$$h_{\text{DCDD}} = h_{\text{RGN}} . \tag{8.12}$$

Combining equations (8.9) and (8.10), we obtain

$$\nu_{\text{DCDD}} = \frac{1}{2}\left(\frac{\|\mathbf{u}^h\|}{\|\mathbf{U}\|}\right)^2 (h_{\text{DCDD}})^2 \| \boldsymbol{\nabla}\|\mathbf{u}^h\| \| . \tag{8.13}$$

9. UGN/RGN-Based Stabilization Parameters for Navier-Stokes Equations

In [4], we proposed to re-define τ_{PSPG} and provided the reason for doing that. We described how we re-define τ_{PSPG} by modifying the definitions of τ_{P3} and τ_{PV3} given by equations (6.15) and (6.19). We proposed to accomplish that by using the expressions

$$\tau_{\text{P3}} = \tau_{\text{P1}} \frac{\|\mathbf{c}\|}{\nu \|\tilde{\mathbf{k}}_{\text{r}}\|} , \quad \tau_{\text{PV3}} = \tau_{\text{PV1}} \frac{\|\mathbf{c}\|}{\nu \|\tilde{\mathbf{k}}_{\text{r}}\|} , \tag{9.1}$$

or the approximations

$$\tau_{\text{P3}} = \tau_{\text{P1}}\, Re \left(\frac{h_{\text{RGN}}}{h_{\text{UGN}}}\right)^2 , \quad \tau_{\text{PV3}} = \tau_{\text{PV1}}\, Re \left(\frac{h_{\text{RGN}}}{h_{\text{UGN}}}\right)^2 . \tag{9.2}$$

In [4], we further stated that these modifications can also be applied to τ_{S3} and τ_{SV3} given by equations (4.10) and (4.14). In [13], we wrote those expressions explicitly as follows:

$$\tau_{\text{S3}} = \tau_{\text{S1}} \frac{\|\mathbf{c}\|}{\nu \|\tilde{\mathbf{k}}_{\text{r}}\|} , \quad \tau_{\text{SV3}} = \tau_{\text{SV1}} \frac{\|\mathbf{c}\|}{\nu \|\tilde{\mathbf{k}}_{\text{r}}\|} , \tag{9.3}$$

$$\tau_{\text{S3}} = \tau_{\text{S1}}\, Re \left(\frac{h_{\text{RGN}}}{h_{\text{UGN}}}\right)^2 , \quad \tau_{\text{SV3}} = \tau_{\text{SV1}}\, Re \left(\frac{h_{\text{RGN}}}{h_{\text{UGN}}}\right)^2 . \tag{9.4}$$

We noted in [13] that if we are dealing with just an advection-diffusion equation, rather than the Navier-Stokes equations of incompressible flows, then the definition of the unit vector $\mathbf{r}$ changes as follows:

$$\mathbf{r} = \frac{\boldsymbol{\nabla}|\phi^h|}{\| \boldsymbol{\nabla}|\phi^h| \|} . \tag{9.5}$$

We also proposed in [13] to re-define τ_{SUGN3} given by equation (7.4) as follows:

$$\tau_{\text{SUGN3}} = \frac{h_{\text{RGN}}^2}{4\nu} \ . \tag{9.6}$$

Furthermore, we proposed in [13] to replace $(\tau_{\text{LSIC}})_{\text{UGN}}$ given by equation (7.7) as follows:

$$(\tau_{\text{LSIC}})_{\text{UGN}} = (\tau_{\text{SUPG}})_{\text{UGN}} \ \|\mathbf{u}^h\|^2 \ . \tag{9.7}$$

We further commented in [13] that the "element length"s h_{UGN} (given by equation (7.1)) and h_{RGN} (equation (8.8)) can be viewed as the local length scales corresponding to the advection- and diffusion-dominated limits, respectively.

10. Deforming-Spatial-Domain/Stabilized Space-Time (DSD/SST) Formulation

In the DSD/SST method, the finite element formulation of the governing equations is written over a sequence of N space-time slabs Q_n, where Q_n is the slice of the space-time domain between the time levels t_n and t_{n+1}. At each time step, the integrations involved in the finite element formulation are performed over Q_n. The space-time finite element interpolation functions are continuous within a space-time slab, but discontinuous from one space-time slab to another. Typically we use first-order polynomials as interpolation functions. The notation $(\cdot)_n^-$ and $(\cdot)_n^+$ denotes the function values at t_n as approached from below and above, respectively. Each Q_n is decomposed into space-time elements Q_n^e, where $e = 1, 2, \ldots, (n_{el})_n$. The subscript n used with n_{el} is to account for the general case in which the number of space-time elements may change from one space-time slab to another. The Dirichlet- and Neumann-type boundary conditions are enforced over $(P_n)_{\text{g}}$ and $(P_n)_{\text{h}}$, the complementary subsets of the lateral boundary of the space-time slab. The finite element trial function spaces $(\mathcal{S}_{\mathbf{u}}^h)_n$ for velocity and $(\mathcal{S}_p^h)_n$ for pressure, and the test function spaces $(\mathcal{V}_{\mathbf{u}}^h)_n$ and $(\mathcal{V}_p^h)_n = (\mathcal{S}_p^h)_n$ are defined by using, over Q_n, first-order polynomials in both space and time.

The DSD/SST formulation is written as follows: given $(\mathbf{u}^h)_n^-$, find $\mathbf{u}^h \in (\mathcal{S}_{\mathbf{u}}^h)_n$ and $p^h \in (\mathcal{S}_p^h)_n$ such that $\forall \mathbf{w}^h \in (\mathcal{V}_{\mathbf{u}}^h)_n$ and $q^h \in (\mathcal{V}_p^h)_n$:

$$\begin{aligned}
&\int_{Q_n} \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{u}^h - \mathbf{f}^h \right) dQ + \int_{Q_n} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(p^h, \mathbf{u}^h) dQ \\
- &\int_{(P_n)_{\text{h}}} \mathbf{w}^h \cdot \mathbf{h}^h dP + \int_{Q_n} q^h \boldsymbol{\nabla} \cdot \mathbf{u}^h dQ + \int_{\Omega_n} (\mathbf{w}^h)_n^+ \cdot \rho \left((\mathbf{u}^h)_n^+ - (\mathbf{u}^h)_n^- \right) d\Omega \\
+ &\sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \frac{\tau_{\text{LSME}}}{\rho} (q^h, \mathbf{w}^h) \cdot \left[(p^h, \mathbf{u}^h) - \rho \mathbf{f}^h \right] dQ \\
+ &\sum_{e=1}^{n_{el}} \int_{Q_n^e} \tau_{\text{LSIC}} \boldsymbol{\nabla} \cdot \mathbf{w}^h \rho \boldsymbol{\nabla} \cdot \mathbf{u}^h dQ = 0,
\end{aligned} \tag{10.1}$$

where

$$(q^h, \mathbf{w}^h) = \rho \left(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{w}^h \right) - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}(q^h, \mathbf{w}^h), \tag{10.2}$$

and τ_{LSME} and τ_{LSIC} are the stabilization parameters (see [14]). This formulation is applied to all space-time slabs $Q_0, Q_1, Q_2, \ldots, Q_{N-1}$, starting with $(\mathbf{u}^h)_0^- = \mathbf{u}_0$. For an earlier, detailed reference on this stabilized formulation see [1].

In [13] we wrote a DSD/SST formulation that was slightly different than the one given by equation (10.1). We did that by neglecting the $(\tau_{\mathrm{LSME}}/\rho)\ \boldsymbol{\nabla} \cdot (2\mu \boldsymbol{\varepsilon}(\mathbf{w}^h))$ term and replacing τ_{LSME} with τ_{SUPG} and τ_{PSPG}:

$$\begin{aligned}
&\int_{Q_n} \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{u}^h - \mathbf{f}^h \right) dQ + \int_{Q_n} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(p^h, \mathbf{u}^h) dQ \\
- &\int_{(P_n)_{\mathrm{h}}} \mathbf{w}^h \cdot \mathbf{h}^h dP + \int_{Q_n} q^h \boldsymbol{\nabla} \cdot \mathbf{u}^h dQ + \int_{\Omega_n} (\mathbf{w}^h)_n^+ \cdot \rho \left((\mathbf{u}^h)_n^+ - (\mathbf{u}^h)_n^- \right) d\Omega \\
+ &\sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \frac{1}{\rho} \left[\tau_{\mathrm{SUPG}} \rho \left(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{w}^h \right) + \tau_{\mathrm{PSPG}} \nabla q^h \right] \cdot \left[(p^h, \mathbf{u}^h) - \rho \mathbf{f}^h \right] dQ \\
+ &\sum_{e=1}^{n_{el}} \int_{Q_n^e} \tau_{\mathrm{LSIC}} \boldsymbol{\nabla} \cdot \mathbf{w}^h \rho \boldsymbol{\nabla} \cdot \mathbf{u}^h dQ = 0.
\end{aligned} \tag{10.3}$$

11. **Element-Matrix-Based Stabilization Parameters for the DSD/SST Formulation**

For extensions of the τ calculations based on matrix norms to the DSD/SST formulation, in [13] we defined the space-time augmented versions of the element-level matrices and vectors given by equations (6.2), (6.6), and (6.10) as follows:

$$\mathbf{c}_{\mathrm{A}} : \qquad \int_{Q_n^e} \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{u}^h \right) dQ \qquad : (\mathbf{c}_{\mathrm{A}})_{\mathrm{V}}, \tag{11.1}$$

$$\tilde{\mathbf{k}}_{\mathrm{A}} : \quad \int_{Q_n^e} \left(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{w}^h \right) \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{u}^h \right) dQ \quad : (\tilde{\mathbf{k}}_{\mathrm{A}})_{\mathrm{V}}, \tag{11.2}$$

$$\gamma_{\mathrm{A}} : \qquad \int_{Q_n^e} \boldsymbol{\nabla} q^h \cdot \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{u}^h \right) dQ \qquad : (\gamma_{\mathrm{A}})_{\mathrm{V}}. \tag{11.3}$$

The components of element-matrix-based τ_{SUPG} were defined in [13] as follows:

$$\tau_{\mathrm{S12}} = \frac{\|\mathbf{c}_{\mathrm{A}}\|}{\|\tilde{\mathbf{k}}_{\mathrm{A}}\|}, \tag{11.4}$$

$$\tau_{\mathrm{S3}} = \tau_{\mathrm{S12}} \frac{\|\mathbf{c}_{\mathrm{A}}\|}{\nu \ \|\tilde{\mathbf{k}}_{\mathrm{r}}\|}, \tag{11.5}$$

where $\tilde{\mathbf{k}}_{\mathrm{r}}$ is the space-time version (i.e. integrated over the space-time element domain Q_n^e) of the element-level matrix given by equation (8.3). To construct τ_{SUPG} from its components we proposed in [13] the form

$$\tau_{\mathrm{SUPG}} = \left(\frac{1}{\tau_{\mathrm{S12}}^r} + \frac{1}{\tau_{\mathrm{S3}}^r} \right)^{-\frac{1}{r}}. \tag{11.6}$$

The components of the element-vector-based τ_{SUPG} were defined in [13] as

$$\tau_{\mathrm{SV12}} = \frac{\|(\mathbf{c}_{\mathrm{A}})_{\mathrm{V}}\|}{\|(\tilde{\mathbf{k}}_{\mathrm{A}})_{\mathrm{V}}\|}, \tag{11.7}$$

$$\tau_{\mathrm{SV3}} = \tau_{\mathrm{SV12}} \frac{\|\mathbf{c}_{\mathrm{A}}\|}{\nu \, \|\tilde{\mathbf{k}}_{\mathrm{r}}\|}. \tag{11.8}$$

From these two components,

$$(\tau_{\mathrm{SUPG}})_{\mathrm{V}} = \left(\frac{1}{\tau_{\mathrm{SV12}}^r} + \frac{1}{\tau_{\mathrm{SV3}}^r} \right)^{-\frac{1}{r}}. \tag{11.9}$$

The components of element-matrix-based τ_{PSPG} were defined in [13] as follows:

$$\tau_{\mathrm{P12}} = \frac{\|\mathbf{g}^{\mathrm{T}}\|}{\|\gamma_{\mathrm{A}}\|}, \tag{11.10}$$

$$\tau_{\mathrm{P3}} = \tau_{\mathrm{P12}} \frac{\|\mathbf{c}_{\mathrm{A}}\|}{\nu \, \|\tilde{\mathbf{k}}_{\mathrm{r}}\|}, \tag{11.11}$$

where $\mathbf{g}^{\mathrm{T}}$ is the space-time version of the element-level matrix given by equation (6.5). To construct τ_{PSPG} from its components, we proposed in [13] the form

$$\tau_{\mathrm{PSPG}} = \left(\frac{1}{\tau_{\mathrm{P12}}^r} + \frac{1}{\tau_{\mathrm{P3}}^r} \right)^{-\frac{1}{r}}. \tag{11.12}$$

The components of the element-vector-based τ_{PSPG} were defined in [13] as follows:

$$\tau_{\mathrm{PV12}} = \frac{\|\mathbf{g}_{\mathrm{V}}^{\mathrm{T}}\|}{\|(\gamma_{\mathrm{A}})_{\mathrm{V}}\|}, \tag{11.13}$$

$$\tau_{\mathrm{PV3}} = \tau_{\mathrm{PV12}} \frac{\|\mathbf{c}_{\mathrm{A}}\|}{\nu \, \|\tilde{\mathbf{k}}_{\mathrm{r}}\|}. \tag{11.14}$$

From these components,

$$(\tau_{\mathrm{PSPG}})_{\mathrm{V}} = \left(\frac{1}{\tau_{\mathrm{PV12}}^r} + \frac{1}{\tau_{\mathrm{PV3}}^r} \right)^{-\frac{1}{r}}. \tag{11.15}$$

The element-matrix-based τ_{LSIC} was defined in [13] as

$$\tau_{\mathrm{LSIC}} = \frac{\|\mathbf{c}_{\mathrm{A}}\|}{\|\mathbf{e}\|}, \tag{11.16}$$

where $\mathbf{e}$ is the space-time version of the element-level matrix given by equation (6.12).

The element-vector-based τ_{LSIC} was defined in [13] as

$$(\tau_{\text{LSIC}})_{\text{V}} = \tau_{\text{LSIC}}. \tag{11.17}$$

12. UGN/RGN-Based Stabilization Parameters for the DSD/SST Formulation

The space-time versions of τ_{SUGN1}, τ_{SUGN2}, τ_{SUGN3}, $(\tau_{\text{SUPG}})_{\text{UGN}}$, $(\tau_{\text{PSPG}})_{\text{UGN}}$, and $(\tau_{\text{LSIC}})_{\text{UGN}}$, given respectively by Eqs. (7.2), (7.3), (9.6), (7.5), (7.6), and (9.7), were defined in [13] as follows:

$$\tau_{\text{SUGN12}} = \left(\sum_{a=1}^{n_{en}} \left| \frac{\partial N_a}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} N_a \right| \right)^{-1}, \tag{12.1}$$

$$\tau_{\text{SUGN3}} = \frac{h_{\text{RGN}}^2}{4\nu}, \tag{12.2}$$

$$(\tau_{\text{SUPG}})_{\text{UGN}} = \left(\frac{1}{\tau_{\text{SUGN13}}^2} + \frac{1}{\tau_{\text{SUGN3}}^2} \right)^{-\frac{1}{2}}, \tag{12.3}$$

$$(\tau_{\text{PSPG}})_{\text{UGN}} = (\tau_{\text{SUPG}})_{\text{UGN}}, \tag{12.4}$$

$$(\tau_{\text{LSIC}})_{\text{UGN}} = (\tau_{\text{SUPG}})_{\text{UGN}} \left\| \mathbf{u}^h \right\|^2. \tag{12.5}$$

Here, n_{en} is the number of nodes for the space-time element, and N_a is the space-time interpolation function associated with node a.

13. Test Computations

13.1. **1D Advection of a Cosine Wave.** In this test, we compute the advection of a cosine wave with the space-time SUPG formulation at dimensionless wave number, $q = kh = 0.3142$ and Courant number, $Cr_u = 1.0$ and 0.5. Figure 1 shows the space-time mesh for $Cr_u = 1.0$ and 0.5.

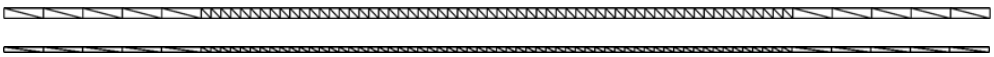


Figure 1. 1D advection of a cosine wave. Meshes for space-time computations. Courant number, $Cr_u = 1.0$ (top) and 0.5 (bottom).

We compare the solutions obtained with the semi-discrete and space-time versions of τ. For the semi-discrete version of τ, we use the expressions given by equations (7.9), (7.3) and (7.5), without τ_{SUGN3}. For the space-time version of τ, we use the expression given by equation (12.1). Figure 2 shows that, for both Courant numbers, the solution obtained with the semi-discrete and space-time versions of τ are very close.

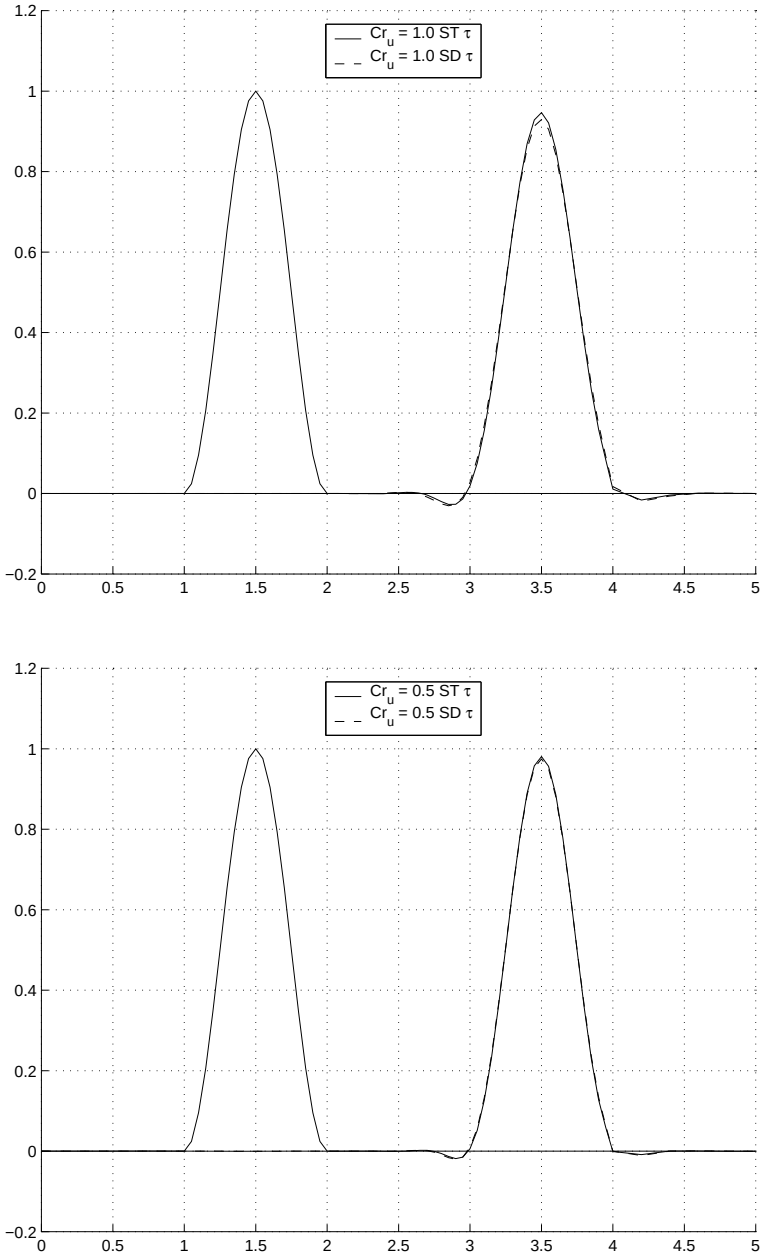


Figure 2. 1D advection of a cosine wave. Stabilized space-time computations with the semi-discrete and space-time versions of τ. Dimensionless wave number, $q = kh = 0.3142$. Courant number, $Cr_u = 1.0$ (top) and 0.5 (bottom).

13.2. **2D Incompressible Flow Past a Cylinder at Re = 100.** In this test computation, for meshes containing elements with high aspect ratios, we evaluate the performance of the SUPG/PSPG formulation with UGN/RGN-based stabilization parameters. The test problem we use, 2D incompressible flow past a cylinder at Re = 100, is a well-studied problem, with an easily identifiable Karman vortex shedding (see Figure 3).

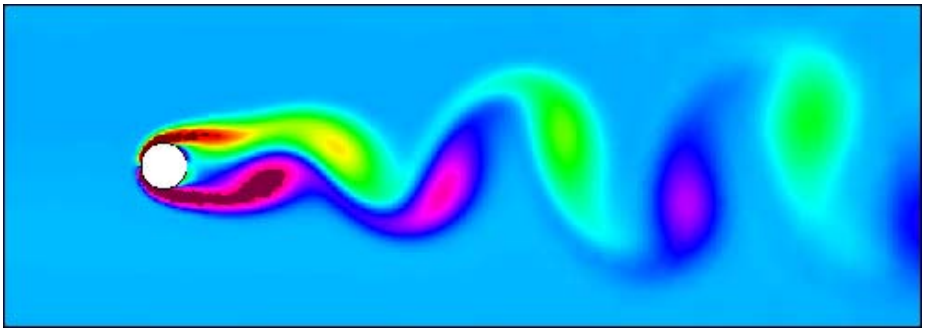


Figure 3. 2D Incompressible Flow Past a Cylinder at Re = 100. Computed with SUPG/PSPG formulation with UGN/RGN-based stabilization parameters. Vorticity.

Figure 4 (4a,b and c) shows the triangular mesh in the boundary layer, the velocity vectors near the cylinder, and the velocity vectors in the boundary layer. Although the aspect ratio of the elements adjacent to the cylinder surface is 100, the SUPG/PSPG formulation with the UGN/RGN-based stabilization parameters performs very well.

14. **Concluding Remarks**

We described how we determine the stabilization parameters ("τ"s) and element length scales used in stabilized finite element formulations of flow problems. These stabilized formulations include the interface-tracking and interface-capturing techniques we developed for computation of flows with moving boundaries and interfaces. The interface-tracking techniques are based on the Deforming-Spatial-Domain/Stabilized Space-Time formulation, where the mesh moves to track the interface. The interface-capturing techniques, typically used with non-moving meshes, are based on a stabilized semi-discrete formulation of the Navier-Stokes equations, combined with a stabilized formulation of an advection equation. The advection equation governs the time-evolution of an interface function marking the interface location. As specific stabilization methods, we focused on the streamline-upwind/Petrov-Galerkin (SUPG) and pressure-stabilizing/Petrov-Galerkin (PSPG) methods. For the Navier-Stokes equations and the advection equation, we described the element-matrix-based and element-vector-based τs designed for semi-discrete and space-time formulations. These τ definitions are expressed in terms of the ratios of the norms of the relevant matrices or vectors. They take into account the local length scales, advection field and the element-level Reynolds number. Based on these definitions, a τ can be calculated

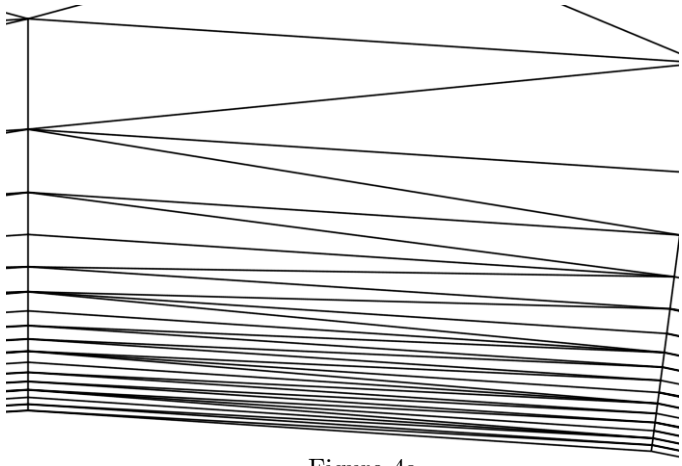


Figure 4a

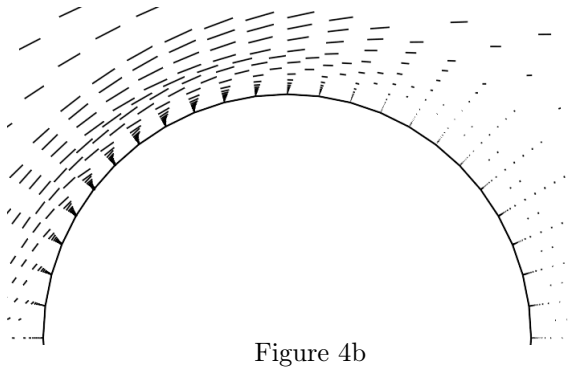


Figure 4b

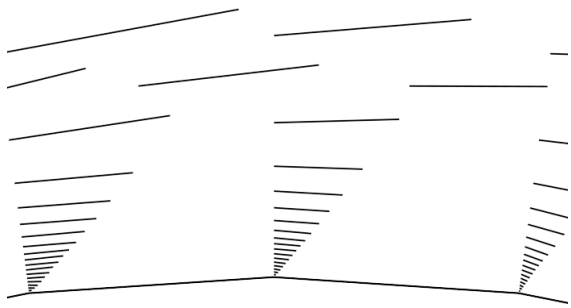


Figure 4c

Figure 4. 2D Incompressible Flow Past a Cylinder at Re = 100. Computed with SUPG/PSPG formulation with UGN/RGN-based stabilization parameters. Top: mesh in the boundary layer; Middle: velocity vectors near the cylinder; Bottom: velocity vectors in the boundary layer.

for each element, or even for each element node or degree of freedom or element equation. We also described certain variations and complements of these new τs, including the approximate versions that are based on the local length scales for the advection- and diffusion-dominated limits. With test problems for the advection-diffusion and Navier-Stokes equations, we showed that the stabilization parameters described perform well, even for elements with high aspect ratios.

References

1. Tezduyar, T.: *Stabilized finite element formulations for incompressible flow computations*, Advances in Applied Mechanics, **28**, (1991), 1–44.
2. Tezduyar, T.: *Finite element methods for flow problems with moving boundaries and interfaces*, Archives of Computational Methods in Engineering, **8**, (2001), 83–130.
3. Tezduyar, T.: *Finite element interface-tracking and interface-capturing techniques for flows with moving boundaries and interfaces*, in *Proceedings of the ASME Symposium on Fluid-Physics and Heat Transfer for Macro- and Micro-Scale Gas-Liquid and Phase-Change Flows (CD-ROM)*, ASME Paper IMECE2001/HTD-24206, ASME, New York, New York, 2001.
4. Tezduyar, T.: *Stabilized finite element formulations and interface-tracking and interface-capturing techniques for incompressible flows* (2002), to appear in *Proceedings of the Workshop on Numerical Simulations of Incompressible Flows*, Half Moon Bay, California.
5. Hughes, T. and Brooks, A.: *A multi-dimensional upwind scheme with no crosswind diffusion*, in T. Hughes, ed., *Finite Element Methods for Convection Dominated Flows*, AMD-Vol.34, ASME, New York (1979), 19–35.
6. Hughes, T., Franca, L. and Hulbert, G.: *A new finite element formulation for computational fluid dynamics: VIII. the Galerkin/least-squares method for advective-diffusive equations*, Computer Methods in Applied Mechanics and Engineering, **73**, (1989), 173–189.
7. Tezduyar, T. and Hughes, T.: *Finite element formulations for convection dominated flows with particular emphasis on the compressible Euler equations*, in *Proceedings of AIAA 21st Aerospace Sciences Meeting*, AIAA Paper 83-0125, Reno, Nevada, 1983.
8. Tezduyar, T. and Park, Y.: *Discontinuity capturing finite element formulations for nonlinear convection-diffusion-reaction problems*, Computer Methods in Applied Mechanics and Engineering, **59**, (1986), 307–325.
9. Tezduyar, T. and Ganjoo, D.: *Petrov-Galerkin formulations with weighting functions dependent upon spatial and temporal discretization: Applications to transient convection-diffusuion problmes*, Computer Methods in Applied Mechanics and Engineering, **59**, (1986), 49–71.
10. L. Franca, L., Frey, S. and Hughes, T.: *Stabilized finite element methods: I. Application to the advective-diffusive model*, Computer Methods in Applied Mechanics and Engineering, **95**, (1992), 253–276.
11. Tezduyar, T. and Osawa, Y.: *Finite element stabilization parameters computed from element matrices and vectors*, Computer Methods in Applied Mechanics and Engineering, **190**, (2000), 411–430.

12. Tezduyar, T.: *Adaptive determination of the finite element stabilization parameters*, in *Proceedings of the ECCOMAS Computational Fluid Dynamics Conference 2001 (CD-ROM)*, Swansea, Wales, United Kingdom, 2001.

13. Tezduyar, T.: *Stabilization parameters and local length scales in SUPG and PSPG formulations* (2002), to appear in *Proceedings of the Fifth World Congress on Computational Mechanics* (Web Site), Vienna, Austria.

14. Tezduyar, T. and Osawa, Y.: *Methods for parallel computation of complex flow problems*, Parallel Computing, **25**, (1999), 2039–2066.

Journal of Computational and Applied Mechanics, Vol. 4., No. 1., (2003), pp. 89–90

Book Review

László P. Kollár, George S. Springer: Mechanics of Composite Structures

An increase in the use of composite materials in many areas of engineering has led to a greater demand for engineers versed in the design of structures made from such materials. Although numerous books offer introductions to composites, few demonstrate advanced concepts or emphasize structures.

This book addresses that need by offering students and engineers tools for designing practical composite structures. The focus is on fiber-reinforced composites composed of fibers embedded in a matrix. Among the topics of interest to the designer are stress-strain relationships for a wide range of anisotropic materials; bending, buckling, and vibration of plates; bending, torsion, buckling, and vibration of solid as well as thin walled beams; shells; hygrothermal stresses and strains; finite element formulation; and failure criteria. The emphasis is on analyses that lead to methods applicable to a variety of structural design problems. The expressions resulting from the analyses are either readily usable or can be translated into a computer algorithm. More than 300 illustrations, 50 fully worked problems, and material properties data sets are included. Some knowledge of composites, differential equations, and matrix algebra is helpful but not necessary, as the book is self-contained.

This book will be of great practical use to graduate students, researchers, and practicing engineers seeking to acquire advanced knowledge of the mechanics of composites and of the applications of composite materials.

***László P. Kollár** is Professor in the Department of Architecture at the Budapest University of Technology and Economics.*

***George S. Springer** is Paul Pigott Professor of Engineering in the Department of Aeronautics and Astronautics at Stanford University.*

***Contents**: Preface; List of symbols; 1. Introduction; 2. Displacements, strains, stresses; 3. Laminated composites; 4. Thin plates; 5. Sandwich plates; 6. Beams; 7. Beams with shear deformation; 8. Shells; 9. Finite element analysis; 10. Failure criteria; 11. Micromechanics; Appendix A. Cross-sectional properties of thin-walled composite beams; Appendix B. Buckling loads and natural frequencies of orthotropic beams with shear deformation; Appendix C. Typical material properties; Index.*

Cambridge University Press

February 2003; 496 pages 339 line diagrams 104 tables 50 exercises; ISBN: 0521801656

Fiber reinforced plastics and composites permit improvements in in-service response and weight reduction compared with more conventional materials. For this reason, the past few decades have seen the rapid expansion of composites in structural applications in many industrial sectors. Examples include applications in automobiles, trains, ships, wind turbine blades, sporting goods, and civil engineering infrastructure such as bridges.

The main advantage of composites over conventional structural materials is that they afford great freedom, not only in selecting the structural forms and dimensions, but also in selecting the appropriate material properties.

To utilize the advantages offered by composites in a wide range of structural applications, empirical and semi-empirical design methods useful in particular cases only must be replaced by methods that are based on fundamental engineering principles and are generally applicable.

The authors of this book set themselves the task of writing a comprehensive treatise that comprised the principles of mechanics and the analysis and design of structural elements made of fiber reinforced composites.

There are engineers engaged in various applications of composites with different professional backgrounds and experience. There are also researchers investigating fundamental aspects of composite materials and composite structures. Hence, it is difficult for the authors to select the content and proper weighing of the different topics so as to satisfy all the readers. And it may be an even more difficult task to find the golden mean between oversimplified and overly theoretical approaches to problems, and between superficial and oversized treatments of topics.

The authors have overcome the aforementioned difficulties. Similarly, due to the different backgrounds and outlooks of the two authors, the book is a uniquely successful synthesis of writing. On the one hand the book is written clearly, with practical users in mind. On the other hand, the text is complete and is based on sound fundamental principles. The former is manifested by the fact that the book is self-contained and by the large number of instructive numerical examples, while the latter is manifested by several new scientific results that have been worked out by the authors for completeness of certain topics. This synthesis also ensured that this comprehensive book has not just become either a collection of formulae or lengthy derivations. It also resulted in numerical examples that not only illustrate solution techniques but shed light on the underlying principles.

István Hegedűs

Journal of Computational and Applied Mechanics, Vol. 4., No. 1., (2003), pp. 91–92

Calendar of Events

2003

August 27-29, 2003: Miskolc, Hungary
9TH Hungarian Conference on Mechanics
Professor G. Stépán, Budapest University of Technology and Economics, Hungary
E-mail: szekeres@mm.bme.hu
Web: http://www.uni-miskolc.hu/mamek-ix/
Phone: +36 1 463-1231; Fax: +36 1 463 3471

September 8-10, 2003: Split, Croatia
BEM 25-25th World Congress on Boundary Element Method
Professor C. A. Brebbia, Wessex Institute of Technology, UK
Professors D. Poljak and V. Roje, University of Split, Croatia
E-mail: rgreen@wessex.ac.uk
Web: http://www.wessex.ac.uk/conferences/2003/bem25/
Phone: 44 (0) 238 029 3223; Fax: 44 (0) 238 029 2853

September 9-12, 2003: Žilina, Slovak Republic
Numerical Methods in Continuum Mechanics
Professor V. Kompiš, University of Žilina, Slovak Republic
E-mail: kompis@mppserv.utc.sk
Web: http://mppserv.utc.sk/NMCM2003
Phone: +421 41 525 3352; Fax: +421 41 5652940

October 22-26, 2003: Taipei, Taiwan
IASS-APCS 2003, International Symposium on New Perspectives
for Shell and Spatial Structures
Dr. C. S. Chen, Secretariat, IASS-APCS 2003
E-mail: cschen@ce.ntu.edu.tw
Web: http://www.caece.net/iass2003
Phone: (886) 2 2367 2558; Fax: (886) 2 2363 1558;

2004

July 5-7, 2004: Cape Town, South Africa
Second International Conference on Structural Engineering,
Mechanics and Computation
Professor A. Zingoni, University of Cape Town, South Africa
E-mail: azingon@eng.uct.ac.za
Web: http://www.ebe.uct.ac.za/~semc2004
Phone: (27) (21) 650 2601; Fax: (27) (21) 689 7471;

August 15-21, 2004: Warsaw, Poland
International Congress of Theoretical and Applied Mechanics
Professor W. Gutkowski, Polish Academy of Sciences
E-mail: ictam04@ippt.gov.pl
Web: http://www.ictam04.ippt.gov.pl
Phone: (+48 22) 826 9803; Fax: (+48 22) 826 9815

Notes for Contributors

to the Journal of Computational and Applied Mechanics

Aims and scope. The aim of the journal is to publish research papers on theoretical and applied mechanics. Special emphasis is given to articles on computational mechanics, continuum mechanics (mechanics of solid bodies, fluid mechanics, heat and mass transfer) and dynamics. Review papers on a research field and materials effective for teaching can also be accepted and are published as review papers or classroom notes. Papers devoted to mathematical problems relevant to mechanics will also be considered.

Frequency of the journal. Two issues a year (approximately 80 pages per issue).

Submission of Manuscripts. Submission of a manuscript implies that the paper has not been published, nor is being considered for publication elsewhere. Papers should be written in standard grammatical English. Two copies of the manuscript should be submitted on pages of A4 size. The text is to be 130 mm wide and 190 mm long and the main text should be typeset in 10pt CMR fonts. Though the length of a paper is not prescribed, authors are encouraged to write concisely. However, short communications or discussions on papers published in the journal must not be longer than 2 pages. Each manuscript should be provided with an English Abstract of about 50–70 words, reporting concisely on the objective and results of the paper. The Abstract is followed by the Mathematical Subject Classification – in case the author (or authors) give the classification codes – then the keywords (no more than five). References should be grouped at the end of the paper in numerical order of appearance. Author's name(s) and initials, paper titles, journal name, volume, issue, year and page numbers should be given for all journals referenced.

The journal prefers the submission of manuscripts in LaTeX. Authors should prefer the $\mathcal{AMS}$-LaTeX article class and are not recommended to define their own LaTeX commands. Visit our home page for further details concerning the issue how to edit your paper.

For the purpose of refereeing, two copies of the manuscripts should initially be submitted in hardcopy to an editor of the journal. The eventual supply of an accepted-for-publication paper in its final camera-ready form (together with the corresponding files on an MS–DOS diskette) will ensure more rapid publication. Format requirements are provided by the home page of the journal from which sample LaTeX files can be downloaded:

`http://www.uni-miskolc.hu/home/web/pumns/mechanics`

These sample files can also be obtained directly (via e-mail) from a member of the Editorial Board, Gy. Szeidl (Gyorgy.SZEIDL@uni–miskolc.hu), upon request.

Twenty offprints of each paper will be provided free of charge and mailed to the correspondent author.

The Journal of Computational and Applied Mechanics is abstracted in Zentralblatt für Mathematik and in the Russian Referativnij Zhurnal.

Responsible for publication: Rector of the Miskolc University
Published by the Miskolc University Press under the leadership of Dr. József PÉTER
Responsible for duplication: works manager Mária KOVÁCS
Number of copies printed: 200
Put to the Press on May 5, 2003
Number of permission: TU 03-400-ME **HU ISSN 1586–2070**

A Short History of the Publications of the University of Miskolc

The University of Miskolc (Hungary) is an important center of research in Central Europe. Its parent university was founded by the Empress Maria Teresia in Selmecbánya (today Banska Štiavnica, Slovakia) in 1735. After the first World War the legal predecessor of the University of Miskolc moved to Sopron (Hungary) where, in 1929, it started the series of university publications with the title *Publications of the Mining and Metallurgical Division of the Hungarian Academy of Mining and Forestry Engineering* (Volumes I.-VI.). From 1934 to 1947 the Institution had the name Faculty of Mining, Metallurgical and Forestry Engineering of the József Nádor University of Technology and Economic Sciences at Sopron. Accordingly, the publications were given the title *Publications of the Mining and Metallurgical Engineering Division* (Volumes VII.-XVI.). For the last volume before 1950 – due to a further change in the name of the Institution – *Technical University, Faculties of Mining, Metallurgical and Forestry Engineering, Publications of the Mining and Metallurgical Divisions* was the title.

For some years after 1950 the Publications were temporarily suspended.

After the foundation of the Mechanical Engineering Faculty in Miskolc in 1949 and the movement of the Sopron Mining and Metallurgical Faculties to Miskolc, the Publications restarted with the general title *Publications of the Technical University of Heavy Industry* in 1955. Four new series - Series A (Mining), Series B (Metallurgy), Series C (Machinery) and Series D (Natural Sciences) - were founded in 1976. These came out both in foreign languages (English, German and Russian) and in Hungarian.

In 1990, right after the foundation of some new faculties, the university was renamed to University of Miskolc. At the same time the structure of the Publications was reorganized so that it could follow the faculty structure. Accordingly three new series were established: Series E (Legal Sciences), Series F (Economic Sciences) and Series G (Humanities and Social Sciences). The latest series, i.e., the series H (European Integration Studies) was founded in 2001. The eight series are formed by some periodicals and such publications which come out with various frequencies.

Papers on computational and applied mechanics were published in the

Publications of the University of Miskolc, Series D, Natural Sciences.

This series was given the name Natural Sciences, Mathematics in 1995. The name change reflects the fact that most of the papers published in the journal are of mathematical nature though papers on mechanics also come out.

The series

Publications of the University of Miskolc, Series C, Fundamental Engineering Sciences

founded in 1995 also published papers on mechanical issues. The present journal, which is published with the support of the Faculty of Mechanical Engineering as a member of the Series C (Machinery), is the legal successor of the above journal.

Journal of Computational and Applied Mechanics

Volume 4, Number 1 (2003)

Contents